CAR-TR-790
CS TR-3527

DAAH04-93-G-0419
September 1995

Electronic Stabilization and Feature Tracking in Long Image Sequences

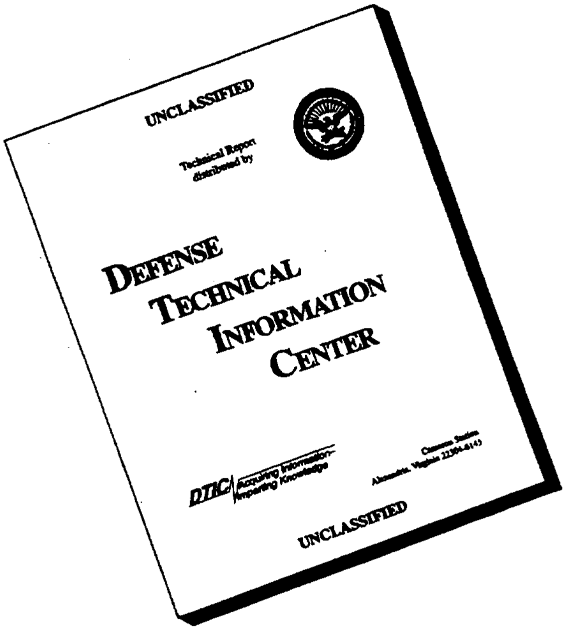
Y.S. Yao
R. Chellappa

Department of Electrical Engineering
and Center for Automation Research
University of Maryland
College Park, MD 20742

COMPUTER VISION LABORATORY

CENTER FOR AUTOMATION RESEARCH

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742-3275



CAR-TR-790
CS-TR-3527

DAAH04-93-G-0419
September 1995

Electronic Stabilization and Feature Tracking in Long Image Sequences

Y.S. Yao
R. Chellappa

Department of Electrical Engineering
and Center for Automation Research
University of Maryland
College Park, MD 20742

Abstract

This dissertation is concerned with processing of visual motion with application to off-road vehicular navigation. Several aspects of the problem are investigated. First, we consider the estimation of total rotation from a sequence. This procedure is important for motion analysis, image stabilization, and independently moving object detection, as well as the recovery of other structural information. We exploit the dynamic nature of a sequence and use multiple visual cues to perform image stabilization. Depending on the knowledge of intrinsic parameters such as the focal length and the field of view, both calibrated and uncalibrated stabilization schemes are designed. The residual motion in a stabilized sequence is also analyzed. Next we address the issue of selective stabilization, defined as the separation of the smooth rotation and the residual oscillatory rotation. In off-road vehicular navigation, in addition to smooth motion, a vehicle exhibits residual vibrations. These residual oscillatory components often affect the interpretation of visual information. We incorporate a kinetic model to explicitly account for vibration phenomena. A maneuver detection scheme, for detecting the beginning and end of smooth rotation, is designed to facilitate the selective stabilization. The structure parameters are consequently estimated in a less perturbed frame of reference. Finally, we study the problem of feature correspondence. Tracking feature points has been a critical procedure in exploiting an image sequence. We propose a localized feature point tracking algorithm. The method employs a 2D kinematic model and relies on a Probabilistic Data Association Filter for the estimation of inter-frame motion. Corresponding points are identified to sub-pixel accuracy and an Extended Kalman Filter is employed to process the new data. The ability to dynamically include new feature points from subsequent frames also makes the algorithm suitable for structure from motion and tracking over a sequence.

The support of the Advanced Research Projects Agency (ARPA Order No. A422) and the Army Research Office under Grant DAAH04-93-G-0419 is gratefully acknowledged, as is the help of Sandy German in preparing this paper.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivations

Vision plays an important role in the successful operation of intelligent mobile robots. In order to interact with its environment, it is essential that the robot have the ability to process dynamic imagery for the recovery of its own motion and relevant environmental structure. Other related tasks such as moving object detection, obstacle avoidance, target acquisition, etc., can be subsequently accomplished.

The interpretation and analysis of visual motion has been an active area of research for over two decades. Depending on the image motion magnitude, either a **continuous** or a **discrete** form is employed to describe image motion; continuous description leads to the computation of **dense** motion fields, while the discrete characterization yields **sparse** feature point displacements. Accordingly, major techniques for motion/structure analysis fall into either the **flow-based** or the **feature-based** domain.

Flow-based methods employ spatial and temporal image gradients to infer motion fields and then seek out the interpretation of motion/structures. Two basic assumptions are involved here. One is the so-called the constant brightness constraint: the image brightnesses of pixels corresponding to the same 3D points remain constant. The other assumption is that the computed optical flow is identical to the motion field—the projection of the 3D velocity vector onto the image plane. However, from the constant brightness constraint, only flow components in the directions of image gradients, i.e. **normal flow**, can be computed directly. Therefore, additional constraints are introduced and regularization techniques are applied for the recovery of optical flow [26, 30, 39]. Techniques employing spatial-temporal filters for the recovery of full flow have also been considered [24]. Using the second assumption, motion and structure are extracted from the computed optical flow [1, 4, 35, 49, 54]. Alternatively, without the explicit computation of optical flow, **direct methods** resorting to the direct estimation of motion/structure [3, 25, 40, 47] and the qualitative interpretation of visual motion [21, 42] have also been pursued. Although great advances have been made in this area, the instability and sensitivity to noise in estimating image derivatives impose limits on flow-based methods.

Feature-based approaches, on the other hand, rely on the **correspondence** of sparse features to recover motion/structure. Depending on the application, different features such as points, lines and contours can be used. Since less stringent constraints than the direct use of the constant brightness constraint are usually employed to obtain the feature displacements, more robust interpretation of motion/structure becomes possible. In addition, much of the early work, the so-called **two-frame based** approaches, focused on using two or three frames in their formulations, resulting in simple linear algorithms [34, 43, 50–52]. However, a lack of robustness of the estimates to correspondence errors has been observed [2, 20, 28]. One way in which robust algorithms can be obtained is by exploiting the dynamic nature of the sequence, i.e. the **long-sequence based** schemes.

Most of the long-sequence based approaches have employed kinematic laws for capturing the dynamic nature of a sequence [7, 46, 56, 58–60]. The motion of a camera is commonly assumed to be smooth in these approaches: in [7], a smooth motion model such as constant translation and constant precision motion is assumed; in [46], the motion is assumed to be constant over the relevant image sequence, in which different types of motion such as pure rotation, dominant translation, etc. are considered; in [56], a Taylor series expansion is employed to describe the trajectory of the rotation center; in [59], a constant acceleration and precision motion model is assumed; in [60] a constant acceleration and rotation model is used, while in [58] a constant translation and rotation kinematic law is adopted. For applications involving indoor surroundings or outdoor environments with flat surfaces, motion may be satisfactorily described by a kinematic law [17]. However, in applications such as off-road vehicular navigation and automatic flight landing [44], the excitation of disturbances results in the deviation of the underlying motion from the assumed kinematic model. Consequently, the interpretation of motion/structure is affected by the presence of these disturbances [18].

We are therefore interested in employing more appropriate dynamic laws for capturing the behavior of mobile robots. Assume that the translation of a vehicle is mainly along its longitudinal axis and its speed can easily be obtained from an odometer, we concentrate on dynamic laws for describing the rotation of a vehicle. The orientation of a vehicle, in general, results from a **desired** smooth rotation and an **undesired** high frequency oscillatory rotation. The smooth rotation arises from the steering command as well as the natural response of a vehicle when it traverses a flat surface. The oscillatory rotation is the **residual** response of mechanical elements to the roughness components in natural terrain. More specifically, the oscillatory rotation is part of the undesired motion which the suspension system is designed to compensate. Although sensors such as gyroscopes and accelerometers can be used to compensate for these effects, this results in a costly system. When dynamic laws are available to describe these phenomena, the separation of both rotational phenomena directly from the imagery becomes possible. We refer to this procedure as **selective stabilization.**

This dissertation therefore deals with various issues related to the robust recovery of motion and structure parameters when a vehicle performs off-road navigation, namely image stabilization, selective stabilization, and tracking of image primitives.

1.1.1 Image Stabilization

Image stabilization, defined here as the warping of video sequences for the removal of image motion due to camera rotation, is a key preprocessing step in dynamic image analysis. This procedure is important for various reasons. The most significant reason is that the rotational flow does not convey structural information and image motion due to camera translation can often be confused with flow resulting from camera rotation [48]. Image stabilization is therefore useful for motion analysis, structure from motion [29, 31], as well as the recovery of the Focus of Expansion (FOE) and other structural information (such as time-to-collision). After performing stabilization, simple and effective independent motion detection mechanisms can be employed to detect independently moving objects [41]. In addition, image stabilization is also important for visual control, as well as for other image exploitation tasks such as registration, object detection [62], automatic target recognition, autonomous vehicle navigation [15], and model-based compression [32].

In this dissertation, we study the use of combined visual cues and dynamical models for the purpose of stabilization, and rely on an Extended Kalman Filter (EKF) for the estimation of the parameters of interest. Since the image motion of features such as distant points and horizon lines is mainly affected by the rotational motion of the camera, both visual cues are exploited in either calibrated or uncalibrated stabilization scheme. For the calibrated scheme where the intrinsic parameters such as the focal length and the intersection of the optical axis on the image plane are known, the imaging model is applied to a 3D vector whose motion characterizes the movement of a horizon line. Consequently, we unify the representations of points and lines for the estimation. For sequences whose intrinsic parameters are unknown, the uncalibrated scheme alternatively exploits the image motion of measurable, 2D normal vectors of horizon lines. Moreover, the study of the sensitivity of relevant parameters with respect to the intrinsic parameters yields similar approaches in both calibrated and uncalibrated schemes. The issues of local and global stabilization schemes are addressed. The residual motion remaining in a stabilized sequence is also analyzed. Finally, both stabilization schemes have been applied to several real sequences, and the results have been made available to interested readers.

1.1.2 Selective Stabilization

As discussed earlier, traditional approaches to the recovery of motion and structure parameters have regarded the effects of high frequency oscillatory motion as noise. As a result, the estimates obtained without the explicit consideration of these disturbances are less desirable for higher level applications such as control and planning.

In this dissertation, we consider both kinematic and kinetic models suitable for capturing different phenomena and achieving the separation of smooth rotation and oscillatory rotation. Our approach uses various dynamic laws to model the behavior of a vehicle, and relies on an EKF for the estimation of both phenomena. In addition to the application of the proposed image stabilization technique, a scheme for detecting the beginning and end of smooth rotation is devised. Appropriate dynamic laws are then employed to achieve selective stabilization. Based on selective stabilization, the 3D locations of close feature points are estimated in a stabilized frame of reference, thus providing more useful information. Synthetic experiments for different scenarios using the proposed approach show promising results.

1.1.3 Feature Tracking

Feature tracking over a sequence is the first step in feature-based approaches for the recovery of motion and structure parameters. Many feature tracking algorithms assume two steps in tracking discrete features. The first step is to detect features using a feature detection scheme. Subsequently, different algorithms employ different methods in matching a set of currently tracked features to the set of features detected in the previous step. The accuracy of these algorithms thereupon relies on the feature detection schemes.

Since detecting the same features in subsequent frames is not easy, we propose a localized feature tracking algorithm. The scheme identifies corresponding points using other techniques to avoid the above problem. The trajectory of each feature point is first described by a 2D kinematic model. Then, to track a feature point, an inter-frame motion estimation scheme is designed to obtain estimates of inter-frame motion parameters. Using these estimates, corresponding points are then identified to sub-pixel accuracy. Temporal information is processed to facilitate subsequent tracking. Since different feature points are tracked independently, the algorithm is able to handle image motion arising from general 3D camera movements. In addition to tracking feature points detected at the beginning, an efficient way to dynamically include new points extracted in subsequent frames is devised so that the information in a sequence is preserved. The experimental results on many real sequences, including the ones used for achieving image stabilization, have shown good performance of the algorithm.

1.2 Contributions

The main contributions of this dissertation are the careful considerations of various issues critical to off-road vehicular navigation. Specific contributions include the following:

- A generic approach exploiting multiple visual cues for image stabilization is proposed.
    - The observability of warping parameters from different image primitives (points and lines at infinity) for both calibrated and uncalibrated cases is studied.
    - The sensitivity of the warping parameters with respect to the intrinsic parameters for uncalibrated cases is investigated.
    - The equivalence of both local and global stabilization schemes is discussed. The nature of the residual motion in the resulting warped sequence is analyzed.
    - Different visual cues are integrated for the estimation of rotational motion in a sequence.
- A scheme which explicitly incorporates the oscillatory behavior of a vehicle when it undergoes off-road navigation is designed.
    - A kinetic law for oscillatory rotation is derived. Its performance is studied both analytically and quantitatively.
    - Separability of smooth rotation and oscillatory rotation is investigated. Selective removal of oscillatory rotation from total rotation is justified.
    - Visual cues are exploited to instantaneously or conservatively detect the beginning and end of smooth rotation.
    - The relative positions of scene points are estimated in a stabilized frame of reference. Impressions of steering and climbing are preserved in the interpretation of visual motion.
- Finally, a localized tracking scheme which performs tracking of a dynamic set of feature points over a sequence is proposed.
    - A localized 2D motion model which can handle the image motion due to general 3D movements of a camera is presented.
    - A scheme for tracking new feature points detected from subsequent frames is designed. The scheme is efficient in the sense that only limited numbers of feature points are tracked at all times.

1.3 Organization

The organization of this dissertation is as follows. Chapter 2 reviews related work on image stabilization and feature tracking. Chapter 3 describes the proposed image stabilization scheme. The recovery of motion/structure from selective stabilization is presented in Chapter 4. Chapter 5 describes the localized feature point tracking algorithm. Chapter 6 concludes this dissertation with discussions of possible future directions.

Chapter 2

Related Work

In this chapter, we review related work on image stabilization and feature tracking.

2.1 Image Stabilization

As mentioned earlier, image stabilization is referred as the process of the removal of unwanted image motion. The unwanted image motion can be due to the total camera motion or only the rotational motion.

When the removal of total image flow is concerned, image stabilization has a similar meaning to image registration. Many techniques for achieving image registration can therefore be applied [8]. Among them, techniques pursued in [10, 23] have attracted a lot of attention. Instead of registering the current image to the previous image, image stabilization is achieved by registering the new frame to a mosaic constructed from previous frames. Specifically, the motion between the current frame and the mosaic is estimated by a multi-resolution, iterative process. During each iteration, cross-correlation based techniques are employed to compute the optical flow between the source image and the warped image. Since the mosaic is used, the overlap between two consecutive images in the source video can therefore be small.

On the other hand, image stabilization is sometimes referred to as the removal of image flow due to rotation. Recently, in addition to approaches to the estimation of 3D motion of a camera, a few promising approaches to this type of image stabilization have been proposed [15, 29, 38, 53].

In [29], assuming that the image region of a planar patch in a 3D scene is detected, the algorithm first computes a quasi-projective transformation which describes the image motion of such a region. Then, by applying the computed transformation to the whole image, the rotation between the current frame and the previous frame is removed. Subsequently, in order to obtain a stabilized sequence as if taken from a camera undergoing translation, the translation of the camera needs to be estimated. The angular velocity of the camera is computed afterwards, and applied to warp the images to achieve image stabilization.

In [15], two methods using normal flow for the recovery of 3D motion parameters are proposed. The first approach uses the pattern of normal flow to estimate the motion of the camera. It is regarded as a scheme for estimating the ego-motion of a camera. The other approach focuses on the estimation of the angular velocity. Since the image motion corresponding to scenes which are far away is mainly due to the camera rotation, this scheme relies on such image regions for the estimation. In order to obtain high accuracy, iterations are performed to refine estimates of the angular velocity using the source frame and the warped image.

In [53], with the help of inertial sensors, image stabilization is achieved by first aligning line segments extracted from an image sequence with the absolute vertical direction. Disparities between two successive frames are then compensated for by 2D linear translations.

Finally, the removal of rotation can also be achieved using image registration. In [38], instead of using quasi-projective transformation to describe the image motion, an affine transformation is employed to model the image flow of distant features. Since the Euclidean distances of features are affected only by the scale between two frames, the scaling factor is first estimated. The remaining 2D translational and rotational parameters are then computed by solving an over-determined system. Since the scheme compensates for the image motion of distant features, it effectively cancels out the rotation of the camera.

2.2 Feature Tracking

Existing techniques for tracking a set of features over a sequence of images generally fall into two categories: two-view based and multiple-view based. Similar to the recovery of 3D motion, when the motion of a camera is smooth such that the smoothness constraints hold, multiple-view based approaches are likely to outperform two-view based methods. On the contrary, if the movement of a camera varies often and result in non-smooth image motion, two-frame based schemes seem to capture the variations more promptly.

For two-view based approaches, finding feature correspondences over a sequence of images is broken into successive, yet independent problems of two-view matching [14, 55, 61]. In [55], multiple attributes of each image point such as intensity, edgeness and cornerness which are invariant under rigid motion in the image plane are used along with a set of constraints to compute a dense displacement field and occlusion areas in two images. An intensity-based cross-correlation method is then used to refine the two-view matching results and obtain feature point correspondences over the sequence [14]. In [61], an image registration technique, based on an affine transformation, is first applied to compensate for the motion of the camera between two consecutive frames. The feature point correspondence problem is then solved by repeatedly identifying the corresponding points to sub-pixel accuracy using a correlation matching method.

Multiple-view based approaches employ smoothness constraints to exploit the temporal information existing in the sequence [6, 11, 45, 60]. In [45], under the assumption that the motion of an object does not change abruptly, the correspondence problem was formulated as an optimization problem. The trajectories of a set of feature points were obtained by searching for a set of trajectories each of which had maximal smoothness. In [6], Multistage Hypothesis Testing (MHT) was applied to detect small, moving objects in each image; a feature trajectory was determined by repeatedly detecting the same feature point over the sequence. In [11], a 2D kinematic motion model was assumed, while the Joint Probabilistic Data Association Filter (JPDAF) was later employed to track line segments with the ability to initiate or terminate the trajectory of a line segment. In [60], assuming 3D kinematic motion model and a Mahalanobis distance based matching criterion, an EKF was used to track a set of line segments. A fading memory type statistical test was suggested to take into account the occlusion and disappearance of line segments.

Chapter 3

3D Model-Based Image Stabilization

3.1 Introduction

Image stabilization is a key preprocessing step in dynamic image analysis and deals with the removal of unwanted image motion in a video sequence. Depending on the final application, this unwanted motion may correspond to part or all of the image flow generated by the 3D camera motion. If the ultimate goal is feedback control of the mechanical system, the undesired image motion corresponds to the residual oscillatory motion of the platform hosting the camera. Image stabilization then has a purpose similar to that of mechanical stabilization. If the task consists of detecting independently moving objects, the unwanted image motion is instead that motion resulting from camera rotation [15]. If the goal is mosaicking and change detection [10, 23], the unwanted image motion corresponds to the total image flow. In this chapter, stabilization is principally understood as the warping of video sequences for the removal of image motion due to total camera rotation. The use of combined visual cues and dynamical models for the stabilization of calibrated or uncalibrated image sequences are described.

Parameters relevant for image warping are estimated by combining information from different tracked tokens, namely points and horizon lines. These parameters are simply the camera rotational velocity if intrinsic camera parameters are available, or the projectivity coefficients, in the uncalibrated case. Image plane displacements of distant feature points may unambiguously characterize rotational motion. However, such points are sometimes difficult to detect and track, due to the absence of sufficient intensity gradient information. For the same reason flow-based methods suffer from a lack of available visual information. Horizon lines, when present, constitute on the other hand very strong visual cues, requiring relatively simple operations for their tracking. These tokens must however be combined so as to remove all ambiguities concerning camera motion. These issues are addressed in the next section.

Subsequently, we address the important issue of the selection of an appropriate dynamic model suitable for exploiting the temporal information in a sequence. In actual applications, cameras are often rigidly mounted on platforms. The rotation of the camera therefore arises from the rotational movement of the host at all times. We evaluate analytically the use of kinetic versus kinematic laws for the estimation of rotational motion components. We discuss the conditions under which the use of simpler kinematic laws yields satisfactory performance.

These analytical results are applied to the stabilization of images acquired from off-road vehicles, for which the rotational motion is significant. Specifically, to stabilize a sequence, horizon lines are first extracted from each frame while distant feature points close to the horizon are detected. Both image primitives are tracked over the sequence. The matched lines and points thereafter form a set of visual cues. Observations are then used along with a kinematic law to estimate the needed warping parameters. Based on the estimated warping parameters, a stabilized sequence is generated.

The organization of this chapter is as follows. Section 3.2 discusses the observability of warping parameters from different visual cues. Residual motion of warped sequences is analyzed in Section 3.3. Section 3.4 compares different dynamics-based estimators for the estimation of rotation. An integrated approach exploiting temporal information as well as multiple visual cues is presented in Section 3.5. Section 3.6 reports experimental results on real image sequences. Conclusions are given in Section 3.7.

3.2 Model-guided Image Warping Schemes

This section addresses the observability of parameters used for image stabilization in both calibrated and uncalibrated cases. We address the recovery of parameters from points and horizon lines. Consider the scenario shown in Figure 3.1 where a camera undergoes rotation with instantaneous angular velocity $\boldsymbol{\omega} : (\omega_x, \omega_y, \omega_z)^T$, and translation with linear velocity $\mathbf{V} : (V_x, V_y, V_z)^T$. Let $\mathbf{P} : (X, Y, Z)^T$ denote the 3D position of a scene point with respect to the camera, and $\mathbf{p} : (x, y)^T$, the image plane coordinates of the corresponding projection point.

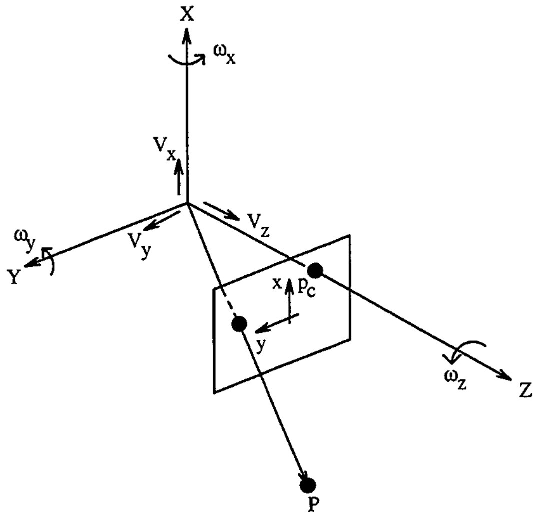


Figure 3.1: The camera motion and imaging model.

The relative motion of the scene point with respect to the camera is then described by

$$\dot{\mathbf{P}} = -\boldsymbol{\omega} \times \mathbf{P} - \mathbf{V} \tag{3.1}$$

Assuming that the perspective projection is used as an imaging model, $\mathbf{p}$ is related to $\mathbf{P}$ as follows:

$$\mathbf{p} = \mathcal{P}(\mathbf{P}) + \mathbf{p}_c \tag{3.2}$$

where $\mathbf{p}_c$ is the intersection of the optical axis with the image plane and $\mathcal{P}$ denotes the perspective projection operator, i.e.

$$\mathcal{P}(\mathbf{P}) = \begin{bmatrix} f_c \frac{X}{Z} \\ f_c \frac{Y}{Z} \end{bmatrix} \tag{3.3}$$

with f_c the focal length. Consequently, the image motion arising from the camera movement satisfies (see Appendix A for derivations)

$$\dot{\mathbf{p}} = \underbrace{[f_c^{-1}(\mathbf{p}-\mathbf{p}_c)\boldsymbol{\omega}_{xy}^{\perp T}(\mathbf{p}-\mathbf{p}_c) + \boldsymbol{\Omega}_z(\mathbf{p}-\mathbf{p}_c) + f_c\boldsymbol{\omega}_{xy}^{\perp}]}_{\dot{\mathbf{p}}_r} + \underbrace{[(\mathbf{p}-\mathbf{p}_{\mathrm{foe}})\frac{1}{\tau}]}_{\dot{\mathbf{p}}_t} \tag{3.4}$$

where $\mathbf{p}_{\mathrm{foe}} = \mathcal{P}(\mathbf{V}) + \mathbf{p}_c$ and $\tau = Z/V_z$ respectively correspond to the Focus of Expansion (FOE) and the time-to-collision to the imaged point; $\boldsymbol{\Omega}_z$ is the 2×2 skew-symmetric matrix related to the rotational velocity component along the optical axis

$$\boldsymbol{\Omega}_z = \begin{bmatrix} 0 & \omega_z \\ -\omega_z & 0 \end{bmatrix}; \tag{3.5}$$

and $\boldsymbol{\omega}_{xy}^{\perp}$ is an image vector orthogonal to the projection of the instantaneous angular velocity component parallel to the image plane,

$$\boldsymbol{\omega}_{xy}^{\perp} = \begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix} \tag{3.6}$$

When image motion is described instantaneously, image velocities $\dot{\mathbf{p}}$ due to 3D rotation of the camera are expressed as second order polynomial functions of image positions and are independent of depth. For this reason, it is a well known fact that image motion resulting from rotation can be instantaneously compensated for. Care has to be taken, however, with regard to the interpretation of the resulting derotated sequence, a point addressed in the next section. On the other hand, unless relative depth is known, or all imaged points lie on the same 3D plane, translational motion cannot be compensated for.

Consider a distant point (i.e. let $\tau \rightarrow \infty$) and denote its position by $\mathbf{P}$. As seen from (3.4) such points move relative to the camera as if only rotation were present. Therefore we may equivalently assume for these points that the 3D motion simply satisfies

$$\dot{\mathbf{P}} = -\boldsymbol{\omega} \times \mathbf{P} \tag{3.7}$$

For off-road vehicle navigation, or images taken from a plane or a helicopter, horizon lines or partial profiles of objects lying far away constitute very strong visual cues. In Figure 3.2, consider an image horizon line denoted by $\mathcal{L}$; $\mathcal{L}$ is uniquely characterized by $\mathbf{W}$, the 3D vector normal to the plane Π through $\mathcal{L}$ and the camera center.

Since the image motion of horizon lines is explained exclusively in terms of the camera rotation, it follows that the motion of the normal vector $\mathbf{W}$ is itself described by

$$\dot{\mathbf{W}} = -\boldsymbol{\omega} \times \mathbf{W} \tag{3.8}$$

If $\mathbf{u}$ is one solution, i.e. $\dot{\mathbf{W}} = -\mathbf{u} \times \mathbf{W}$, then $k\mathrm{W} + \mathbf{u}$ also satisfies (3.8) for any k. Therefore, observing the image motion of one horizon line characterizes the rotational component on plane Π only. There is indeterminacy along the direction $\mathbf{W}$. Given one observation, the set of possible solutions of (3.8) describes an affine line in 3D rotational parameter space. When only one horizon line is observed, rotation may be determined in the Least Square (LS) sense which corresponds to the camera motion with least energy that explains the image motion of that particular line. In this case rotational motion inducing image motion along the line feature itself (lateral motion) is not always totally compensated for (since it is assumed to be zero), and other lines or geometrical cues

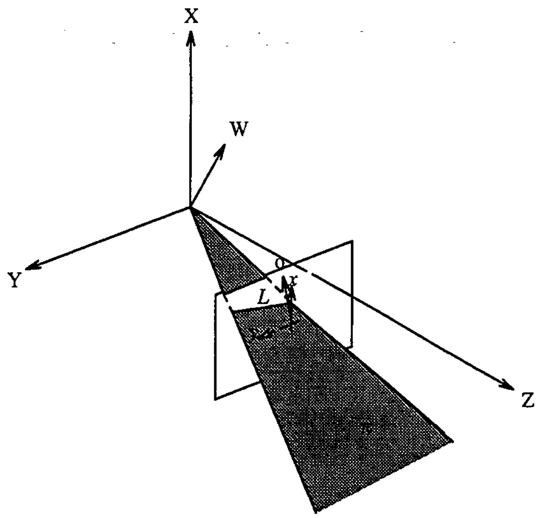


Figure 3.2: Geometric representation of a horizon line $\mathcal{L}$, the plane Π, the 3D normal vector $\mathbf{W}$ and the image plane normal vector $\mathbf{o}$.

near the horizon can be used to qualify lateral motion for full stabilization. Indeterminacy exists also if only one distant point is observed. In this case, this indeterminacy involves the rotational component along a ray from the image center to this point. Qualitatively, however, lines and points carry equal amounts of information in the determination of rotation. If any combination of two of these features is observed, rotation can be fully characterized, except in some degenerate cases for which $\mathbf{W} \times \mathbf{P} = \mathbf{0}$, which in practice cannot occur unless the observer possesses an unreasonably large Field of View (FOV). Letting $\mathbf{w} = \mathcal{P}(\mathbf{W}) + \mathbf{p}_c$, one may solve the over-determined linear system

$$\boldsymbol{Q\omega = D} \tag{3.9}$$

where $\mathbf{D} = [\dot{\mathbf{p}}_1^T, \ldots, \dot{\mathbf{p}}_M^T \dot{\mathbf{w}}_1^T, \ldots, \dot{\mathbf{w}}_N^T]^T$, while $\mathbf{Q}$ is a matrix derived from (3.4):

$$\mathbf{Q} = \begin{bmatrix} (x_1 - x_c)(y_1 - y_c) & -(f_c + (x_1 - x_c)^2) & (y_1 - y_c) \\ (f_c + f_c^{-1}(y_1 - y_c)^2) & -f_c^{-1}(x_1 - x_c)(y_1 - y_c) & -(x_1 - x_c) \\ \vdots & \vdots & \vdots \\ (x_M - x_c)(y_M - y_c) & -(f_c + (x_M - x_c)^2) & (y_M - y_c) \\ (f_c + f_c^{-1}(y_M - y_c)^2) & -f_c^{-1}(x_M - x_c)(y_M - y_c) & -(x_M - x_c) \\ (\mathrm{w}_{x_1} - x_c)(\mathrm{w}_{y_1} - y_c) & -(f_c + (\mathrm{w}_{x_1} - x_c)^2) & (\mathrm{w}_{y_1} - y_c) \\ (f_c + f_c^{-1}(\mathrm{w}_{y_1} - y_c)^2) & -f_c^{-1}(\mathrm{w}_{x_1} - x_c)(\mathrm{w}_{y_1} - y_c) & -(\mathrm{w}_{x_1} - x_c) \\ \vdots & \vdots & \vdots \\ (f_c + f_c^{-1}(\mathrm{w}_{y_N} - y_c)^2) & -f_c^{-1}(\mathrm{w}_{x_N} - x_c)(\mathrm{w}_{y_N} - y_c) & -(\mathrm{w}_{x_N} - x_c) \end{bmatrix} \tag{3.10}$$

Line features can therefore be combined with other tracked tokens such as distant points for stabilization.

We now discuss the image warping schemes. Irrespective of the particular trajectories of the vectors $\boldsymbol{\omega}(t)$ and $\mathbf{V}(t)$, the positions of 3D points at two time instants can always be described by an element of the Special Euclidean group $SE(3)$ (uniquely, if the rotation center is given), i.e., there exist a total rotation $\mathbf{R} : [r_{ij}], i = 1, \ldots, 3, j = 1, \ldots, 3$ and translation $\mathbf{T} : (T_x, T_y, T_z)^T$, between any two frames, such that the 3D point positions $\mathbf{P}_1$ and $\mathbf{P}_2$ expressed in the camera frame of reference satisfy

$$\mathbf{P}_2 = \mathbf{R}\mathbf{P}_1 + \mathbf{T} \tag{3.11}$$

As before, for a distant point, the contribution from the translation $\mathbf{T}$ is negligible. The image plane positions of such a point at t_1 and t_2 are then expressed as

$$\mathbf{p}_2 = (\mathbf{c}^T\mathbf{p}_1 + 1)^{-1}(\mathbf{A}\mathbf{p}_1 + \mathbf{b}) \tag{3.12}$$

where

$$\begin{aligned}
\mathbf{A} &= d^{-1}\begin{bmatrix} f_c r_{11} + x_c r_{31} & f_c r_{12} + x_c r_{32} \\ f_c r_{21} + y_c r_{31} & f_c r_{22} + y_c r_{32} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} && (3.13)\\
\mathbf{b} &= d^{-1}\begin{bmatrix} -f_c x_c r_{11} - f_c y_c r_{12} + f_c^2 r_{13} - x_c^2 r_{31} - x_c y_c r_{32} + f_c x_c r_{33} \\ -f_c x_c r_{21} - f_c y_c r_{22} + f_c^2 r_{23} - x_c y_c r_{31} - y_c^2 r_{32} + f_c y_c r_{33} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} && (3.14)\\
\mathbf{c} &= d^{-1}\begin{bmatrix} r_{31} \\ r_{32} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} && (3.15)\\
d &= -x_c r_{31} - y_c r_{32} + f_c r_{33} && (3.16)
\end{aligned}$$

Similarly, for a distant line feature characterized by the projected normal vector $\mathbf{w}$,

$$\mathbf{w}_2 = (\mathbf{c}^T\mathbf{w}_1 + 1)^{-1}(\mathbf{A}\mathbf{w}_1 + \mathbf{b}) \tag{3.17}$$

As seen from (3.12) and (3.17), for distant features, their image plane motions are described by projective group operations. In fact, this is expected since points on the horizon fall into a plane at infinity and the image motion of planes is described exactly by a projectivity [1]. Consequently, if $\{r_{i,j}, i = 1,\ldots,3, j = 1,\ldots,3\}$ in $\mathbf{A}, \mathbf{b}$ and $\mathbf{c}$ are estimated from points and lines near the horizon, we can compensate for the rotation between two images using the transformation

$$\begin{aligned}
\mathbf{p}_{c2} &= (\mathbf{p}_2\mathbf{c}^T - \mathbf{A})^{-1}(\mathbf{b} - \mathbf{p}_2) && (3.18)\\
&= (\tilde{\mathbf{c}}^T\mathbf{p}_2 + 1)^{-1}(\tilde{\mathbf{A}}\mathbf{p}_2 + \tilde{\mathbf{b}}) && (3.19)
\end{aligned}$$

where $\mathbf{p}_{c2}$ represents the points on the compensated image. This is also a projective transformation with parameters $\tilde{\mathbf{A}}, \tilde{\mathbf{b}}$ and $\tilde{\mathbf{c}}$, as expected by virtue of the projective group property. It is shown later that there exists only a translation between stabilized images. Therefore, (3.19) forms the basis of our stabilization scheme for calibrated sequences.

Note that simpler image transformations have been used in the literature for stabilization purposes: the *SE(2)* group of transformation and affine transformations, i.e. $\mathbf{p}_{c2} = \tilde{\mathbf{A}}\mathbf{p}_2 + \tilde{\mathbf{b}}$, are some examples used in works such as [15, 29];[1] these are essentially appropriate in cases of parallel-frontal motion.

In cases where the intrinsic parameters are unknown, we show that it is still possible to achieve image stabilization using both distant points and lines. For points, since $\mathbf{p}$ is directly measurable from the images, the projective transformation in (3.12) remains unchanged. However, due to the unknown $\mathbf{p}_c$ and f_c, the mapping for distant lines in (3.17) is no longer applicable since $\mathbf{w}$ is no longer measurable. Consider instead the measured image normal vector to a line near the horizon described in the image plane by

$$\mathbf{o}^T\mathbf{p} = 1 \tag{3.20}$$

It is shown in Appendix B that $\mathbf{o}$ is related to the previously defined projected normal $\mathbf{w}$ by

$$\mathbf{w} = -(1- < \mathbf{p}_c, \mathbf{o} >)^{-1} f_c^2 \mathbf{o} + \mathbf{p}_c \tag{3.21}$$

[1] In [29], a second order polynomial quasi-projective transformation is assumed, $\mathbf{p}_{ct} = \mathbf{C}\mathbf{p}\mathbf{p}^T\mathbf{d} + \mathbf{A}\mathbf{p} + \mathbf{b}$; an affine transformation is then used for further motion analysis.

Using the identities $\mathbf{o}_2^T\mathbf{p}_2 = 1$ and $\mathbf{o}_1^T\mathbf{p}_1 = 1$ along with (3.12) we can further show that the movement of $\mathbf{o}$ also satisfies a projective transformation whose parameters are related to the inverse projective transformation $\tilde{\mathbf{A}}$, $\tilde{\mathbf{b}}$, $\tilde{\mathbf{c}}$ in (3.19):

$$\mathbf{o}_2 = (-\tilde{\mathbf{b}}^T\mathbf{o}_1 + 1)^{-1}(\tilde{\mathbf{A}}^T\mathbf{o}_1 - \tilde{\mathbf{c}}) \tag{3.22}$$

Consequently, image stabilization of uncalibrated sequences can be carried out by estimating the eight parameters in $\mathbf{A}, \mathbf{b}$ and $\mathbf{c}$ from distant points and lines. While this is possible in principle, in practice, the computation of the projectivity parameters from distant features can be unstable. To see this, consider the system obtained from (3.12),

$$\mathbf{Gf} = \mathbf{q} \tag{3.23}$$

where $\mathbf{f}$ is the vector including the eight projectivity parameters, $\mathbf{f} = (a_{11}, \ldots, c_2)^T$, $\mathbf{q}$ consists of the coordinates of a set of points such as $\mathbf{p}_2$, and $\mathbf{G}$ is a matrix with elements composed of the coordinates of $\mathbf{p}_1$ and $\mathbf{p}_2$. This matrix is often ill conditioned; it can be shown that the last two columns of $\mathbf{G}$ contain second order terms in the image coordinates, while the third and sixth columns contain zeroth order terms. Geometrically, since the eight parameters are computed from features lying only on a constrained region of the image (namely close to the horizon), there exist many projectivities leaving the horizon invariant, some of which are not suitable. One possible solution to this problem is to further constrain the intrinsic parameters by assuming an approximate value and concentrating on estimation of the rotation.

Indeed, there often are situations where the intrinsic parameters are approximately known.[2] In these cases, denote the intrinsic parameters by $\boldsymbol{\lambda} = (f_c, x_c, y_c)^T$, and let the nominal values be $\boldsymbol{\lambda}_0$. If we further assume that the eight projective coefficients vary smoothly with respect to $\boldsymbol{\lambda}$ then

$$\begin{aligned}\mathbf{f}(\boldsymbol{\lambda}) &\approx \mathbf{f}(\boldsymbol{\lambda}_0) + \left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\lambda}}\right|_{\boldsymbol{\lambda}_0}(\boldsymbol{\lambda} - \boldsymbol{\lambda}_0) & (3.24)\\ &= \mathbf{f}(\boldsymbol{\lambda}_0) + \mathbf{J}(\boldsymbol{\lambda}_0)(\boldsymbol{\lambda} - \boldsymbol{\lambda}_0) & (3.25)\end{aligned}$$

where $\mathbf{J}$ is the Jacobian matrix. When the elements of $\mathbf{J}$ are small, the effect of the imperfect knowledge of the intrinsic parameters is negligible for stabilization purposes. A small error in the assumed intrinsic parameters will still lead to acceptable stabilization results. For example, consider the sensitivity of b_1 with respect to f_c. Then from (3.14) and assume the optical axis intersects the image plane at the center, i.e. $(x_c, y_c) = (0, 0)$, we have

$$\begin{aligned}\frac{\partial b_1}{\partial f_c} &= r_{13} \cdot r_{33}^{-1} & (3.26)\\ &\approx -\omega_y & (3.27)\end{aligned}$$

where (3.27) is typically true since the rotation between two consecutive frames is small, say on the order of 10^{-3}. Consequently, if the error in f_c is within 100 pixels, the error in b_1 will be within one pixel. Since similar arguments can be applied to other coefficients, we can concentrate on estimating the three rotational parameters as in the case of calibrated sequences. It will be seen later that this assumption holds in real applications.

[2] In fact, the errors in these parameters can be moderate, as shown in the experiments later.

3.3 Residual Motion Analysis

We analyze here the nature of the resulting sequence when the image warping described in the previous section is applied. We further compare two types of stabilization: local stabilization and global stabilization. For convenience, denote the sequence of original images by $\mathcal{S} = \{I_0, I_1, \ldots, I_n\}$ and call the compensated sequence $\mathcal{S}_c = \{I_0, I_{c1}, \ldots, I_{cn}\}$. To obtain a translation-only sequence $\mathcal{S}_c$, either a local or global stabilization scheme can be utilized. The local approach directly compensates for the rotation between I_{k+1} and I_{ck} to generate I_{ck+1}, while the global method computes the rotation between I_{k+1} and I_k.

Consider first the residual motion in a sequence generated by a local stabilization scheme. Since every frame I_{k+1} is directly stabilized with respect to the previous compensated image I_{ck}, we have

$$\mathbf{P}_{ck+1} = \mathbf{R}^{c^{-1}}_{k+1,k}\mathbf{P}_{k+1} \tag{3.28}$$

where $\mathbf{R}^{c}_{k+1,k}$ denotes the rotation between I_{ck} and I_{k+1}, while $\mathbf{P}_{k+1}$ and $\mathbf{P}_{ck+1}$ respectively represent the 3D coordinates of a scene point relative to the camera coordinate system in I_{k+1} and I_{ck+1}. Also, recall that the motion of $\mathbf{P}_k$ in the original sequence is described by

$$\mathbf{P}_{k+1} = \mathbf{R}_{k+1,k}\mathbf{P}_k + \mathbf{T}_{k+1,k} \tag{3.29}$$

with $\mathbf{R}_{k+1,k}$ and $\mathbf{T}_{k+1,k}$ the rotation and translation between the camera frames of reference in I_k and I_{k+1}. Substituting (3.29) into (3.28) and using (3.28) to relate $\mathbf{P}_{ck}$ and $\mathbf{P}_k$, the residual motion between I_{ck+1} and I_{ck} can be expressed as follows:

$$\mathbf{P}_{ck+1} = \mathbf{P}_{ck} + \mathbf{R}^{c^{-1}}_{k+1,k}\mathbf{T}_{k+1,k} \tag{3.30}$$

Furthermore, $\mathbf{R}^{c}_{k+1,k}$ is related to the rotation in the original sequence by

$$\begin{aligned} \mathbf{R}^{c}_{k+1,k} &= \mathbf{R}_{k+1,k}\mathbf{R}^{c}_{k,k-1} \\ &= \cdots \\ &= \mathbf{R}_{k+1,k}\mathbf{R}_{k,k-1}\cdots\mathbf{R}_{1,0} \end{aligned} \tag{3.31}$$

Therefore, as expected, the compensated sequence exhibits a purely translational motion. If $\mathbf{R}^{c}_{k+1,k}$ is known, the compensated sequence can be used for ego-motion recovery. Observe, however, that while the magnitude of the translation in the stabilized sequence is the same as that of the translation magnitude in the original sequence, the apparent translational heading (and therefore the FOE) now rotates with the original rotational motion.

This local stabilization scheme, however, is not practical for real applications. As seen from (3.31), $\mathbf{R}^{c}_{k+1,k}$ in fact accounts for the rotation between I_0 and I_{k+1}. For large k, the motion between I_0 and I_{k+1} is likely to be large. The overlap between I_{ck} and I_{k+1} may therefore be small. The common features are more difficult to find and consequently, it is not easy to compute $\mathbf{R}^{c}_{k+1,k}$ reliably. Global stabilization, on the other hand, focuses on the estimation of the rotation between I_k and I_{k+1}; the stabilized image is created afterwards with respect to the reference frame, say I_0, according to

$$\mathbf{P}_{ck+1} = (\mathbf{R}_{k+1,k}\mathbf{R}_{k,k-1}\cdots\mathbf{R}_{1,0})^{-1}\mathbf{P}_{k+1} \tag{3.32}$$

The sequence generated by the global stabilization scheme therefore exhibits the same residual motion as the sequence obtained using the local stabilization approach. However, in contrast with the local approach, the global stabilization scheme is more likely to provide reliable estimates of the parameters of interest in real applications, since the area of overlap between I_k and I_{k+1} is greater. This scheme is therefore employed in our work.

3.4 Exploitation of Dynamics for Full Stabilization

The next issue involves the robust estimation of the warping parameters used in the stabilization scheme presented above. As argued earlier, the inherent nature of stabilization implies that temporal information present in the sequence should be exploited for image warping purposes. A relevant problem then becomes that of selecting suitable parameter dynamics so as to exploit this temporal information. We concentrate on the dynamics for the rotation parameters. The incorporation of the proposed dynamic laws into the estimation process is presented later in Section 3.5.

We consider here dynamic laws appropriate for capturing the evolution of rotational motion of the platform on which the camera is mounted. Since the movement of the platform is affected by the interaction between mechanical elements and its environment, the motion is best described by a kinetic model which emulates the behavior of the mechanical system. The use of simpler kinematic models is compared to that of more complete kinetic models.

This section therefore starts with the description of a kinetic model of a vehicle. The presentation of this model leads to the study of selective stabilization, which will be addressed in the next chapter. Subsequently, using this model, the appropriateness of kinematic laws for describing the evolution of rotational parameters is evaluated.

3.4.1 Kinetic Models

As discussed earlier, the movement of a vehicle over rough terrain, in general, can be decomposed into two components: the smooth motion and the residual oscillatory motion. The smooth motion corresponds to the behavior of the vehicle as if the terrain were smooth; it includes translation, as well as rotation due to steering and climbing. The residual oscillatory motion refers to the residual vehicular motion; it characterizes the response of the vehicle to the roughness of the terrain.

We proceed by employing a four-wheel vehicle model to account for the residual oscillatory vehicular movement. This model takes into account the phenomena of bounce, pitch and roll (illustrated in Figure 3.3), and has been widely used for the design and analysis of suspension systems. All tires are modeled by linear springs with the same stiffness coefficient K_T. M_{wf} and M_{wr} represent the masses of unsprung elements such as the front and rear wheels and their axles. K_f, C_f, K_r and C_r are the characteristics of the linear springs and shock absorbers modeling the suspension system.

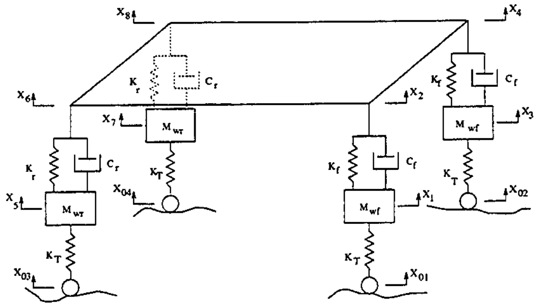


Figure 3.3: The four-wheel vehicle model [22]

Assume that each tire contacts the terrain at a point at all times, there exist seven degrees of freedom: the movements of unsprung elements $\{x_1, x_3, x_5, x_7\}$, the bouncing displacement of the center of gravity of the sprung element x_c, the pitch angle θ and the roll angle ϕ (illustrated in

Figure 3.4, in which W_A and W_B constitute the wheel base, while $T_{s,r}$ and $T_{s,l}$ are the suspension tracks). The high frequency yaw motion is usually small during driving; it can therefore be neglected. Note that because of the decomposition of the vehicle's movement, these oscillatory states are measured with respect to the equilibrium positions resulting from the smooth motion. Their behavior is described in detail in Appendix C. Consequently, if we define $\mathbf{x}_{us}$ to be the vector containing the residual rotational components,

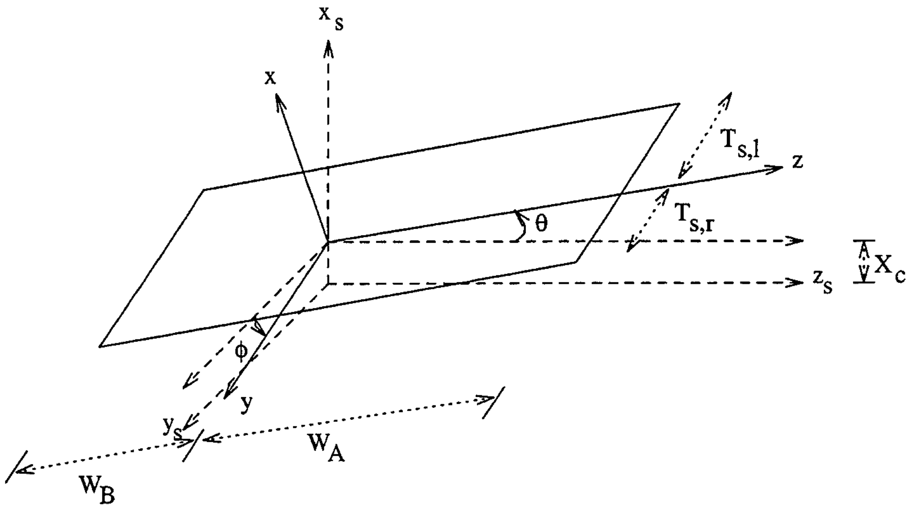


Figure 3.4: The residual oscillatory motion of the four-wheel vehicle model: the bounce x_c, the pitch angle θ, and the roll angle ϕ.

$$\mathbf{x}_{us} \equiv (\theta, \dot{\theta}, \phi, \dot{\phi})^T \tag{3.33}$$

then $\mathbf{x}_{us}$ satisfies

$$\dot{\mathbf{x}}_{us} = \mathbf{\Phi}_{us}\mathbf{x}_{us} + \mathbf{\Gamma}_{us}\mathbf{u}_{us} \tag{3.34}$$

where $\mathbf{u}_{us}$ includes other dependent quantities,

$$\mathbf{u}_{us} \equiv (\mathbf{x}_b^T, \mathbf{x}_w^T)^T \tag{3.35}$$

with $\mathbf{x}_b = (x_c, \dot{x}_c)^T$ and $\mathbf{x}_w = (x_1, \dot{x}_1, x_3, \dot{x}_3, x_5, \dot{x}_5, x_7, \dot{x}_7)^T$. $\mathbf{\Phi}_{us}$ and $\mathbf{\Gamma}_{us}$ respectively denote constant matrices whose entries are related to system parameters such as the mass of the sprung element, the moment of inertia, the spring characteristics, etc.

3.4.2 Rotation Dynamics

We turn now to the evaluation of kinetic models (as compared with simple kinematic models) to capture the evolution of rotational motion. The simple models are sometimes casually used without further examination. We show analytically that kinetic models are superior to kinematic models, but that kinematic models yield acceptable performance if sufficient number of visual observations are available.

As mentioned previously, the attitude change between two time instants is due to smooth rotation and the residual oscillatory motion. Since the smooth rotation due to climbing and banking lasts only for a short period, we study the performance of a kinematic-law based estimator for the estimation of pitch and roll motion only. The estimator can easily be generalized to take into account all the rotational components (including the smooth motion components).

For simplicity, let the measurements $\mathbf{z}$ available for the estimation of $\mathbf{x}_{us}$ are related to $\mathbf{x}_{us}$ by a linear transformation $\mathbf{C}$,

$$\mathbf{z} = \mathbf{C}\mathbf{x}_{us} \tag{3.36}$$

Then assuming that the residual oscillatory rotational motion component is exactly described by the dynamics of $\mathbf{x}_{us}$, we compare the performance of estimators based on different dynamics when additional measurements are available. The kinetic-law based estimator $\hat{\mathbf{y}}_1$ captures the oscillatory motion using coefficients related to the true dynamics [12]:

$$\dot{\hat{\mathbf{y}}}_1 = \mathbf{\Phi}_{us}\hat{\mathbf{y}}_1 + \mathbf{K}_1[\mathbf{z} - \mathbf{C}\hat{\mathbf{y}}_1] \tag{3.37}$$

where $\hat{\mathbf{y}}_1$ represents the resulting estimate of $\mathbf{x}_{us}$, and $\mathbf{K}_1$ is the desired gain matrix. The kinematic-law based estimator $\hat{\mathbf{y}}_2$, on the other hand, assumes no knowledge of the underlying mechanical system and therefore only employs the smooth variation dynamics

$$\dot{\hat{\mathbf{y}}}_2 = \mathbf{\Phi}\hat{\mathbf{y}}_2 + \mathbf{K}_2[\mathbf{z} - \mathbf{C}\hat{\mathbf{y}}_2] \tag{3.38}$$

where $\hat{\mathbf{y}}_2$ and $\mathbf{K}_2$ respectively have meanings similar to those of $\hat{\mathbf{y}}_1$ and $\mathbf{K}_1$, while

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_{01} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}_{01} \end{bmatrix} \tag{3.39}$$

with $\mathbf{0}$ the 2×2 zero matrix, and

$$\mathbf{\Phi}_{01} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{3.40}$$

Let us define the corresponding estimation errors as

$$\tilde{\mathbf{y}}_1 = \mathbf{x}_{us} - \hat{\mathbf{y}}_1 \tag{3.41}$$

$$\tilde{\mathbf{y}}_2 = \mathbf{x}_{us} - \hat{\mathbf{y}}_2 \tag{3.42}$$

It can be shown that $\tilde{\mathbf{y}}_1$ and $\tilde{\mathbf{y}}_2$ satisfy

$$\dot{\tilde{\mathbf{y}}}_1 = (\mathbf{\Phi}_{us} - \mathbf{K}_1\mathbf{C})\tilde{\mathbf{y}}_1 + \mathbf{\Gamma}_{us}\mathbf{u}_{us} \tag{3.43}$$

$$\dot{\tilde{\mathbf{y}}}_2 = (\mathbf{\Phi} - \mathbf{K}_2\mathbf{C})\tilde{\mathbf{y}}_2 + \mathbf{\Gamma}_{us}\mathbf{x}_{us} + \Delta\mathbf{\Phi}_{us}\hat{\mathbf{y}}_2 \tag{3.44}$$

where $\Delta\mathbf{\Phi}_{us} = \mathbf{\Phi}_{us} - \mathbf{\Phi}$ is the mismatch between the assumed and true dynamics. If $\mathbf{K}_1$ is chosen so that the eigenvalues of $\mathbf{\Phi}_{us} - \mathbf{K}_1\mathbf{C}$ have negative real parts, then $\tilde{\mathbf{y}}_1$ remains bounded as long as $\mathbf{u}$ is bounded. Similarly, if $\Delta\mathbf{\Phi}_{us}\hat{\mathbf{y}}_2$ is, in addition, bounded, we can choose $\mathbf{K}_2$ so that $\tilde{\mathbf{y}}_2$ is bounded. Nonetheless, $\tilde{\mathbf{y}}_2$ will exhibit a larger error than $\tilde{\mathbf{y}}_1$.

The kinetic-law estimator, however, requires knowledge of the mechanical system parameters. These parameters are not always easily measurable. When the system parameters are unknown, the availability of sufficient number of visual measurements should allow for the use of the simpler kinematic law while still yielding good warping parameter estimates. The next section employs the kinematic laws for the estimation of total rotation.

3.5 Parameter Estimation

Various types of estimators can be employed to obtain the parameters used in our stabilization scheme. Recursive-type estimators update the estimates of the parameters whenever new information becomes available while batch-type estimators compute the estimates by processing all the information. Because of their ability to process data sequentially and their lower computational complexity, recursive-type estimators are preferred over batch estimators for real time processing. Kalman filters make use of dynamics for their estimation. We therefore provide Extended Kalman Filter (EKF) formulations for both the calibrated and uncalibrated stabilization cases.

For both calibrated and uncalibrated sequences, the algorithm only needs to estimate three rotational parameters, and the state vector $\mathbf{x}$ is simply equal to

$$\mathbf{x} = \boldsymbol{\omega} \tag{3.45}$$

Based on the simple kinematic law justified in the previous section, we have

$$\dot{\mathbf{x}} = 0 \tag{3.46}$$

Subsequently,

$$\mathbf{x}(t_{i+1}) = \mathbf{x}(t_i) \tag{3.47}$$

This constitutes the plant equation in our recursive estimation algorithm. We turn next to the observation equations. Assume that the tracked tokens are composed of M points (whose projection points are $\mathbf{p}_1, \ldots, \mathbf{p}_M$) and N horizon lines (each one with associated normal vector $\mathbf{w}_i, i = 1, \ldots, N$ for calibrated sequences, or alternatively with given $\mathbf{o}_i, i = 1, \ldots, N$ for uncalibrated sequences, with $\mathbf{o}_i$ defined as in (3.20)). The measurement vector in the calibrated case is defined by

$$\mathbf{z} = (\mathbf{p}_1^T, \ldots, \mathbf{p}_M^T, \mathbf{w}_1^T, \ldots, \mathbf{w}_N^T)^T \tag{3.48}$$

From (3.12) and (3.17) we can write the measurement equations as

$$\mathbf{z}(t_{i+1}) = \mathbf{h}_{i+1,i}[\mathbf{x}(t_{i+1})] + \mathbf{n}(t_{i+1}) \tag{3.49}$$

where $\mathbf{h}$ is a nonlinear function while $\mathbf{n}$ denotes the measurement noise. More specifically, in the calibrated case, $\mathbf{h}_{i+1,i}[\mathbf{x}(t_{i+1})]$ is expressed as

$$\mathbf{h}_{i+1,i}[\mathbf{x}(t_{i+1})] = \begin{bmatrix} (\mathbf{c}^T\mathbf{p}_1 + 1)^{-1}(\mathbf{A}\mathbf{p}_1 + \mathbf{b}) \\ \vdots \\ (\mathbf{c}^T\mathbf{p}_M + 1)^{-1}(\mathbf{A}\mathbf{p}_M + \mathbf{b}) \\ (\mathbf{c}^T\mathbf{w}_1 + 1)^{-1}(\mathbf{A}\mathbf{w}_1 + \mathbf{b}) \\ \vdots \\ (\mathbf{c}^T\mathbf{w}_N + 1)^{-1}(\mathbf{A}\mathbf{w}_N + \mathbf{b}) \end{bmatrix} \tag{3.50}$$

with $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$ expressed with respect to the state vector components as

$$\mathbf{A} = d^{-1}\begin{bmatrix} f_c r_{11} + x_c r_{31} & f_c r_{12} + x_c r_{32} \\ f_c r_{21} + y_c r_{31} & f_c r_{22} + y_c r_{32} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{3.51}$$

$$\mathbf{b} = d^{-1}\begin{bmatrix} -f_c x_c r_{11} - f_c y_c r_{12} + f_c^2 r_{13} - x_c^2 r_{31} - x_c y_c r_{32} + f_c x_c r_{33} \\ -f_c x_c r_{21} - f_c y_c r_{22} + f_c^2 r_{23} - x_c y_c r_{31} - y_c^2 r_{32} + f_c y_c r_{33} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{3.52}$$

$$\mathbf{c} = d^{-1}\begin{bmatrix} r_{31} \\ r_{32} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{3.53}$$

$$d = -x_c r_{31} - y_c r_{32} + f_c r_{33} \tag{3.54}$$

The measurement vector in the uncalibrated case is instead defined by

$$\mathbf{z} = (\mathbf{p}_1^T, \ldots, \mathbf{p}_M^T, \mathbf{o}_1^T, \ldots, \mathbf{o}_N^T)^T \tag{3.55}$$

and the observation equation then uses

$$\mathbf{h}_{i+1,i}[\mathbf{x}(t_{i+1})] = \begin{bmatrix} (\mathbf{c}^T\mathbf{p}_1 + 1)^{-1}(\mathbf{A}\mathbf{p}_1 + \mathbf{b}) \\ \vdots \\ (\mathbf{c}^T\mathbf{p}_M + 1)^{-1}(\mathbf{A}\mathbf{p}_M + \mathbf{b}) \\ (-\tilde{\mathbf{b}}^T\mathbf{o}_1 + 1)^{-1}(\tilde{\mathbf{A}}^T\mathbf{o}_1 - \tilde{\mathbf{c}}) \\ \vdots \\ (-\tilde{\mathbf{b}}^T\mathbf{o}_N + 1)^{-1}(\tilde{\mathbf{A}}^T\mathbf{o}_N - \tilde{\mathbf{c}}) \end{bmatrix} \tag{3.56}$$

with $\tilde{\mathbf{A}}$, $\tilde{\mathbf{b}}$, and $\tilde{\mathbf{c}}$ derived from the identities

$$\tilde{\mathbf{A}}^T = \tilde{d}^{-1}\begin{bmatrix} a_{22} - b_2c_2 & b_2c_1 - a_{21} \\ b_1c_2 - a_{12} & a_{11} - b_1c_1 \end{bmatrix} \tag{3.57}$$

$$-\tilde{\mathbf{c}} = \tilde{d}^{-1}\begin{bmatrix} a_{22}c_1 - a_{21}c_2 \\ a_{11}c_2 - a_{12}c_1 \end{bmatrix} \tag{3.58}$$

$$-\tilde{\mathbf{b}} = \tilde{d}^{-1}\begin{bmatrix} a_{22}b_1 - a_{12}b_2 \\ a_{11}b_2 - a_{21}b_1 \end{bmatrix} \tag{3.59}$$

$$\tilde{d} = a_{11}a_{22} - a_{12}a_{21} \tag{3.60}$$

With the plant and measurement equations given in (3.47) and (3.49), when horizon lines and points are tracked, the EKF scheme can be applied to recursively estimate the two-frame angular velocity. This scheme consists of the following steps:

- **Step 1:** State and covariance propagation

$$\begin{aligned} \hat{\mathbf{x}}(t_{i+1}^-) &= \hat{\mathbf{x}}(t_i^+) \\ \Sigma(t_{i+1}^-) &= \Sigma(t_i^+) + \Sigma_w(t_{i+1}) \end{aligned} \tag{3.61}$$

where $\hat{\mathbf{x}}(t_i^+)$ and $\Sigma(t_i^+)$ denote the estimates of $\mathbf{x}(t_i)$ and the associated covariances: they are obtained based on information contained in the sequence up to the i^{th} frame. $\hat{\mathbf{x}}(t_{i+1}^-)$ and $\Sigma(t_{i+1}^-)$, on the other hand, are the predicted estimates of $\mathbf{x}(t_{i+1})$ and the predicted covariances respectively before the incorporation of the $(i+1)^{\text{th}}$ frame, while $\Sigma_w(t_{i+1})$ is the covariance of the plant noise $\mathbf{w}(t_{i+1})$.

- **Step 2:** State and covariance update

$$\begin{aligned} \mathbf{K}(t_{i+1}) &= \Sigma(t_{i+1}^-)\mathbf{H}_{i+1,i}^T[\mathbf{H}_{i+1,i}\Sigma(t_{i+1}^-)\mathbf{H}_{i+1,i}^T + \Sigma_n(t_{i+1})]^{-1} \\ \hat{\mathbf{x}}(t_{i+1}^+) &= \hat{\mathbf{x}}(t_{i+1}^-) + \mathbf{K}(t_{i+1})\left\{\mathbf{z}(t_{i+1}) - \mathbf{h}_{i+1,i}[\hat{\mathbf{x}}(t_{i+1}^-)]\right\} \\ \Sigma(t_{i+1}^+) &= [\mathbf{I} - \mathbf{K}(t_{i+1})\mathbf{H}_{i+1,i}]\Sigma(t_{i+1}^-) \end{aligned} \tag{3.62}$$

where $\hat{\mathbf{x}}(t_{i+1}^+)$ is the desired estimate of $\mathbf{x}(t_{i+1})$, and $\Sigma(t_{i+1}^+)$ is the associated covariance. $\mathbf{K}(t_{i+1}), \Sigma_n(t_{i+1})$ and $\mathbf{I}$ respectively represent the gain matrix, the covariance of $\mathbf{n}(t_{i+1})$ and the identity matrix. $\mathbf{H}_{i+1,i}$, on the other hand, is the linearized approximation of $\mathbf{h}_{i+1,i}$,

$$\mathbf{H}_{i+1,i} = \left.\frac{\partial \mathbf{h}_{i+1,i}}{\partial \mathbf{x}(\mathbf{t}_{\mathbf{i}+1})}\right|_{\hat{\mathbf{x}}(t_{i+1}^-)} \tag{3.63}$$

The algorithm is initialized by a batch process using an LS estimate of the rotational parameters as in

$$\min_{\mathbf{x}(t_1)} \sum_i ||\mathbf{z}(t_{i+1}) - \mathbf{h}_{i+1,i}[\mathbf{x}(t_{i+1})]||^2 \tag{3.64}$$

The minimization can be solved using techniques based on gradient descent, such as conjugate descent.

This concludes our discussion of parameter estimation. The next section addresses image primitive tracking. Synthetic and real experiments are reported as well.

3.6 Experimental Results

We first briefly describe approaches to the detection and tracking of horizon lines and points. Experimental results on two real image sequences are then presented.

3.6.1 Detection/Tracking of Image Primitives

The first set of visual cues for characterizing the rotation consists of horizon lines. There have been numerous approaches to tracking a set of line segments over a sequence [11, 60]. For simplicity, we only focus on tracking one line in our work, although using a set of lines is desirable in some situations.

Assuming that the lines near the horizon appear in the form of large vertical brightness derivatives, the detection of the line of interest is achieved in three steps:

- **Step 1**: Based on the histogram of the image, a binary image is generated by thresholding the original image.
- **Step 2**: In the binary image, the pixels with large vertical brightness derivatives most likely constitute points on the horizon line. A simple template matching technique is applied to identify pixels with large brightness derivatives.
- **Step 3**: A median fit method is used to robustly group identified pixels into lines. The longest line is taken to be the line of interest.

In the experiments, the above procedures are applied to each image in the sequence. The line extracted from each image is assumed to be near the horizon, and these lines are matched to each other.

The second class of inputs to the full stabilization scheme are the image plane trajectories of a set of points near the horizon. The localized tracking algorithm described in Chapter 5 is employed to obtain feature point trajectories and is not repeated here. However, in our formulation, it is not required that the same set of points be used. Therefore, we only need to focus on finding matching points between two frames. A different set of points can be used whenever this is desirable.

3.6.2 Synthetic Experiments

This section tests the results of Section 3.4; we illustrate the performance of the kinematic law using a one-dimensional example, and show that both laws yield comparable performance when sufficient number of visual observations are present. Consider the scenario in which the vehicle travels along a straight path most of the time except when encountering impulse-like disturbances (bumps). Each bump is modeled by a half sine wave with a given height and width. (For simplicity, no smooth rotational motion during driving is assumed.) The excitations to the vehicle are then given by

$$x_{01}(t) = \begin{cases} b_h \sin[\frac{\pi}{b_w}(vt - z_0)] & \frac{z_0}{v} \leq t \leq \frac{z_0+b_w}{v} \\ 0 & \text{elsewhere} \end{cases}$$

with

$$x_{02}(t) = x_{01}(t), \quad x_{03}(t) = x_{01}\left(t - \frac{L}{v}\right), \quad x_{04}(t) = x_{02}\left(t - \frac{L}{v}\right)$$

where z_0, L, and v respectively denote the location of the bump, the vehicle's length and its forward speed. (Only one bump is used in this example, with $Z_0 = 1.345\,\text{m}$, $b_h = 0.1\,\text{m}$, $b_w = 0.2\,\text{m}$, $L = 2.7$ m and $v = 13.45\ m \cdot \text{s}^{-1}$). The nominal values of the vehicle parameters are listed in Table 3.1, and the oscillatory motion is synthesized according to the behavior of the model, derived in Appendix C, with only the pitching motion shown in Figures 3.5a and 3.5b.

During driving, a sequence of images is acquired at 20 Hz from an on-board camera (each image has size 2.0×2.0 and resolution 2000×2000), and a set of distant points is tracked; the coordinates of these points with respect to the initial camera frame of reference are listed in Table 3.2.

Table 3.1: Model parameters. M_B: the mass of the sprung element, I_{yy}: the moment of inertia along the pitch axis, W_A and W_B: the distance of the center of gravity to the front and rear ends.

M_B	I_{yy}	M_{wf}	M_{wr}	K_T
1710.0kg	1031.3kg·m^2	57.5kg	75.0kg	200.0kN·m^{-1}
K_f	K_r	C_f (C_r)	W_A	W_B
18.0kN·m^{-1}	10.0kN·m^{-1}	1.0kN·m^{-1}·s^{-1}	1.4m	1.3m

Table 3.2: Distant point locations.

Point	3D coordinates		
1	84	-84	1200
2	160	-80	2000
3	170	115	2300
4	100	140	1750

Subsequently, to compare the kinetic and and kinematic law based estimators, we design a recursive-type estimator similar to the one used in the real application to estimate the pitching motion. The resulting bias and Root-Mean-Squared Error (RMSE) in the estimates of the attitude change between t and $t + \Delta t$, $\theta_d(t) \equiv \theta(t) - \theta(t - \Delta t)$, are shown in Figures 3.5c and 3.5d. As seen from the figures, when reliable image cues are available, the performance of the kinematic-law based estimator (dashed line) is very close to that of the kinetic-law based one (solid line). More importantly, the larger bias in the kinematic-law based estimator, due to the modeling error in the time evolution of the parameters, is negligible.

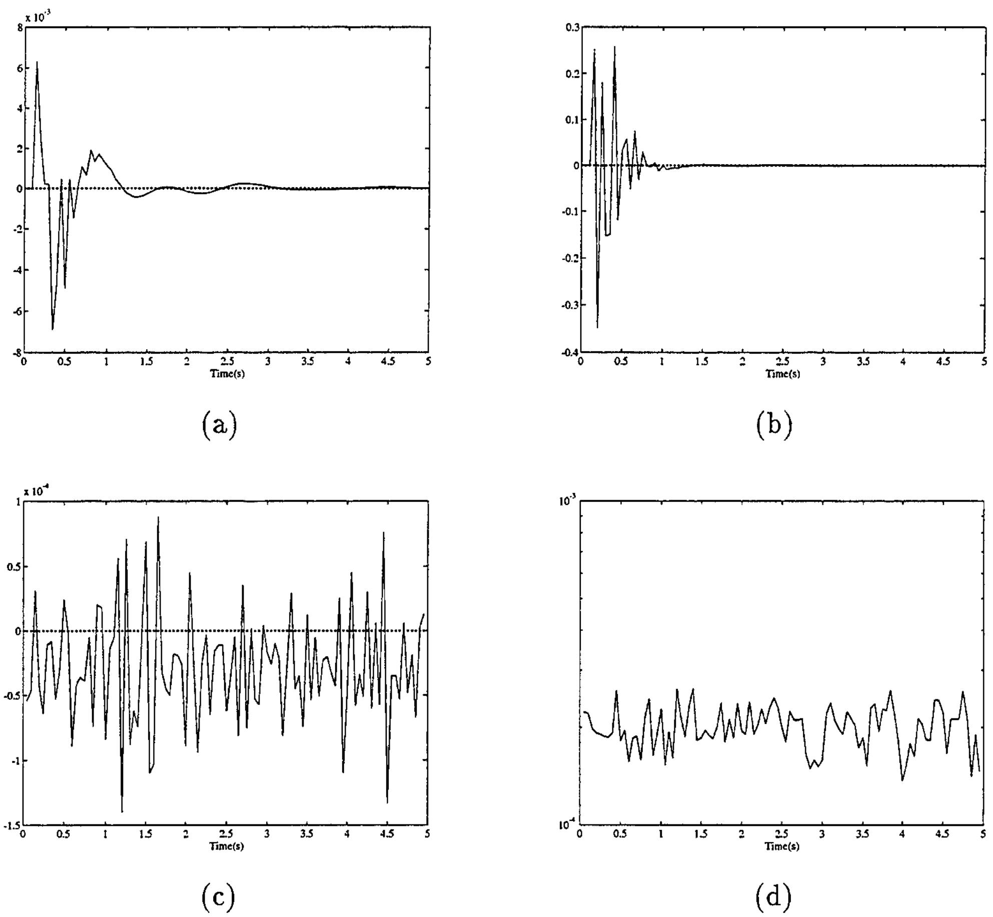


Figure 3.5: Kinetic versus kinematic: (a) $\theta(t)$, (b) $\dot{\theta}(t)$, (c) Bias in the estimates of $\theta_d(t)$, (d) RMSE in the estimates of $\theta_d(t)$.

3.6.3 Real Imagery

We show the results of the application of the stabilization schemes to two sequences with significant unstabilized components in both calibrated and uncalibrated cases. To let the reader more precisely judge the results yielded by this technique, we have placed MPEG movies showing the original sequences, the primitive tracking and the stabilized sequences at the following URL:

http://www.cfar.umd.edu/˜yao/stabilization.html.

They can easily be viewed using any web-browser (Mosaic, NetScape, etc.).

Martin Marietta Sequence The first sequence, distributed by Martin Marietta, was obtained from a vehicle performing off-road navigation. Figure 3.6 shows four frames from the sequence; each frame has size 347 × 238. The motion is composed of translation (with a dominant looming component) and unstabilized rotation. The FOV is 40 × 30 degrees (in the horizontal and vertical directions, respectively), while the optical axis intersects the image plane at the center of the image. The experiments were first carried out for the calibrated case.

In this sequence, both the mountain profile and the line near the horizon are detected. Fig-

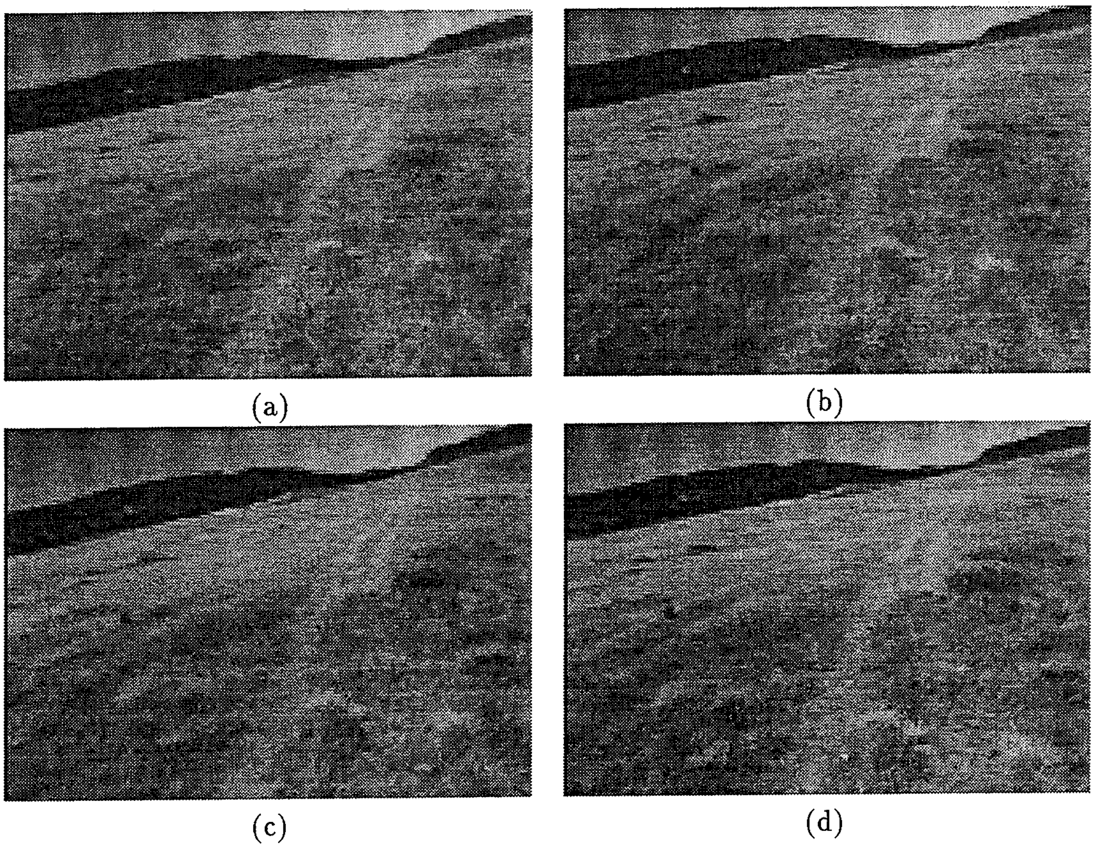


Figure 3.6: Martin Marietta sequence: (a) Image 9, (b) Image 39, (c) Image 74, (d) Image 99.

ure 3.7a shows the results for one frame. (The mountain profile is not employed by the current algorithm.) Four feature points close to the detected line are identified and tracked over the sequence afterwards. For display purposes, the resulting image plane trajectories are superimposed on the last frame of the sequence as shown in Figure 3.7b. The LS estimate of the rotation is also computed as if only the horizon line were available. Subsequently, stabilization from both full and LS rotation is applied; the recursive algorithm, in addition, uses the estimates from the batch algorithm computed from the first ten frames. Figures 3.7c and 3.7d respectively show the LS and full angular velocity estimates. The solid line, the dashed line and the dashed-dotted line respectively correspond to pitch motion, yaw motion and roll motion respectively. Since the vehicle exhibits very small lateral motion and the sequence is densely sampled, the LS estimates are therefore close to the fully stabilized estimates for this sequence. Finally, a stabilized sequence is obtained using the global stabilization scheme with the full rotational estimates. The stabilized sequence appears to undergo translation only, with the direction of translation varying noticeably when the rotational component in the original sequence is large.

The stabilization scheme for the uncalibrated case was also applied, assuming that the intrinsic parameters are unknown. The sensitivity of the eight parameters with respect to the four intrinsic parameters, obtained using the calibrated parameters as nominal values, are first shown in Figures 3.8 and 3.9.[3] (The additional parameter is due to the different focal length in either direction.) For evaluation purposes, we only illustrate the sensitivity of each parameter with respect to

[3] Instead of showing the computed $\mathbf{J}$, Figures 3.8 and 3.9 show the *relative sensitivity*, defined as $\mathbf{J} \cdot \mathbf{\Lambda}$, with $\mathbf{\Lambda}$ a diagonal matrix whose elements are λ_0.

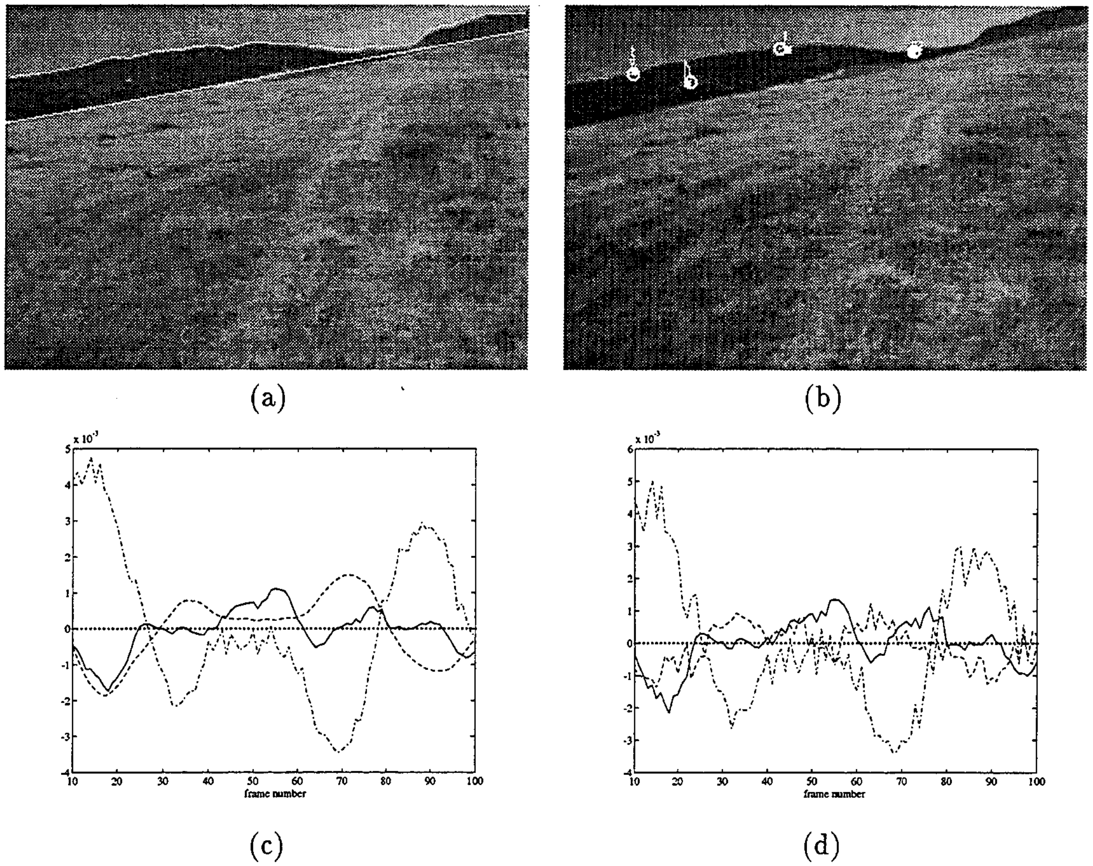


Figure 3.7: Experimental results from the calibrated scheme for Martin Marietta sequence: (a) The detected line and the mountain profile, (b) Feature point trajectories, (c) LS estimates, (d) Fully stabilized estimates.

the focal length in the horizontal direction and x_c.

As discussed earlier, these values are small and therefore a reasonable deviation from the nominal values of the intrinsic parameters is not critical. Consequently, we vary the parameters and perform different tests. For example, assume the focal length in pixels to be 700 and 600 in the horizontal and vertical directions respectively (the real values being 480 and 443 respectively), and the image center to be 50 pixels away from the true position, say (100, 100). Then using the same visual cues, Figure 3.10 shows the estimated angular velocity. The uncalibrated stabilization scheme is thereafter applied to the original sequence. It can be seen from the MPEG movie that the result is close to the one obtained using the calibrated scheme. The result for the uncalibrated sequence using unconstrained projectivity parameters, i.e. direct estimation of the eight parameters from the image features, leads instead to very poor results, since the estimation is unstable.

NIST Sequence The same procedure wass applied to another sequence, provided by NIST, which was also acquired from an off-road vehicle. Figure 3.11 shows four frames of the sequence in which each frame is of size 640 × 480. The motion is composed of a translation with steering and unstabilized motion components. The FOV is 70 and 60 degrees along the horizontal and vertical directions respectively, while the optical axis again intersects the image plane at the center of each image.

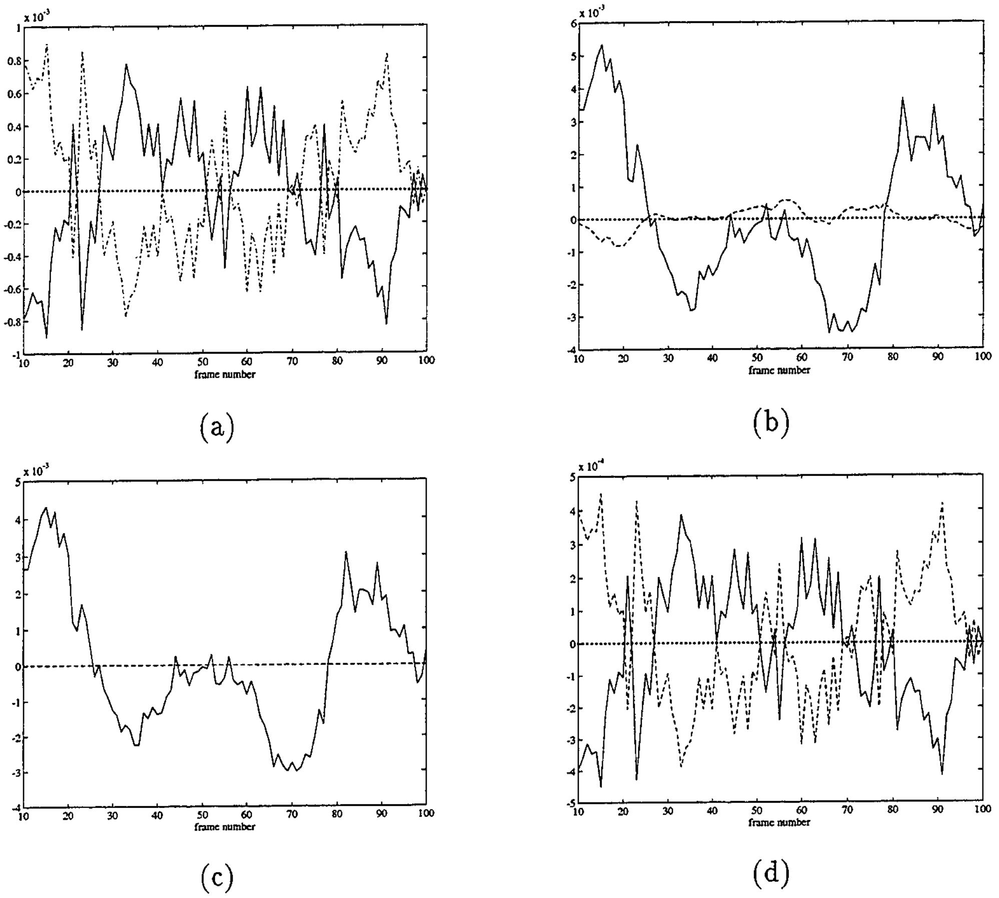


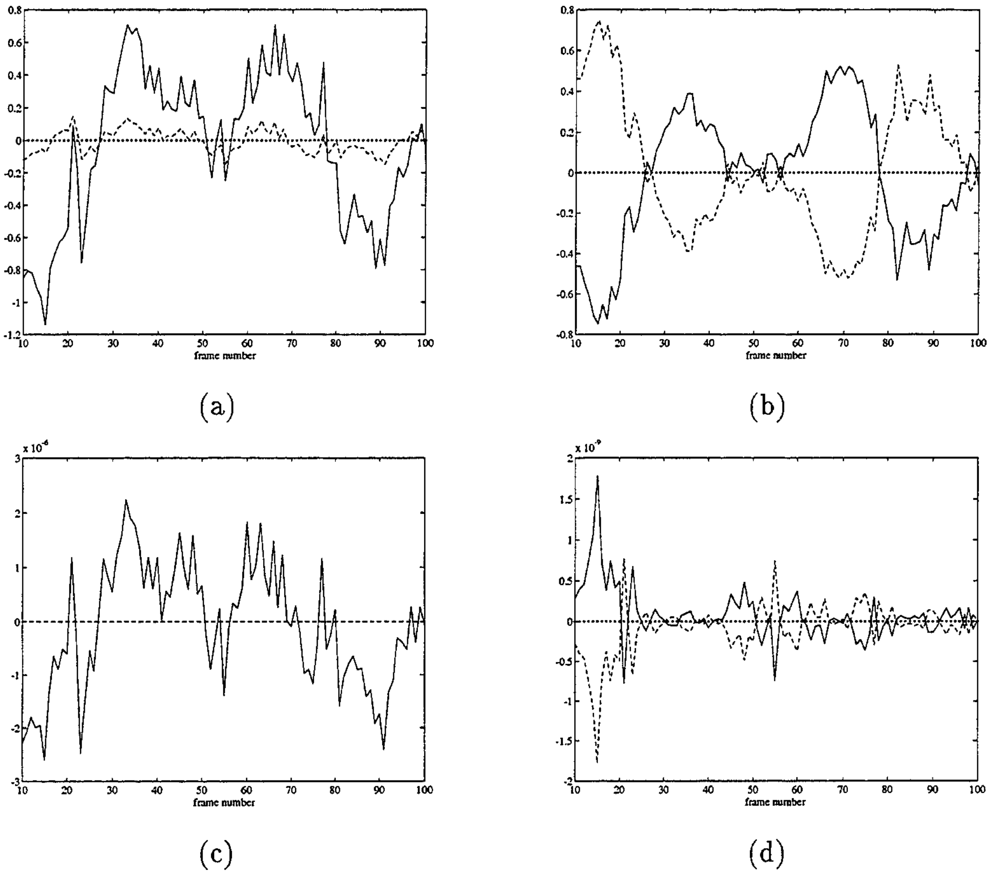
Figure 3.8: Sensitivities of **A** with respect to f_c in the horizontal direction (solid line) and x_c (dashed line) for Martin Marietta sequence: (a) a_{11}, (b) a_{12}, (c) a_{21}, (d) a_{22}.

A line and partial object profiles near the horizon are detected as displayed in Figure 3.12a, while the feature trajectories are similarly superimposed on the last frame as shown in Figure 3.12b. The LS and full estimates of the angular velocity are plotted in a similar fashion in Figures 3.12c and 3.12d respectively. Since the sequence exhibits some steering, the LS and full estimates are quite different. In addition, the steering behavior is preserved in the stabilized sequence; only the unstabilized motion is removed.

We also applied the uncalibrated stabilization scheme to this sequence. Again, we choose the focal length in pixels to be 600 and 700 (in contrast with the true values 249 and 457), and the center of the image to be (250, 300). The sensitivity of the projective coefficients is shown in Figures 3.13 and 3.14. The angular velocity estimates are illustrated in Figure 3.15. The MPEG movie again shows good stabilization results for this sequence.

3.7 Conclusion

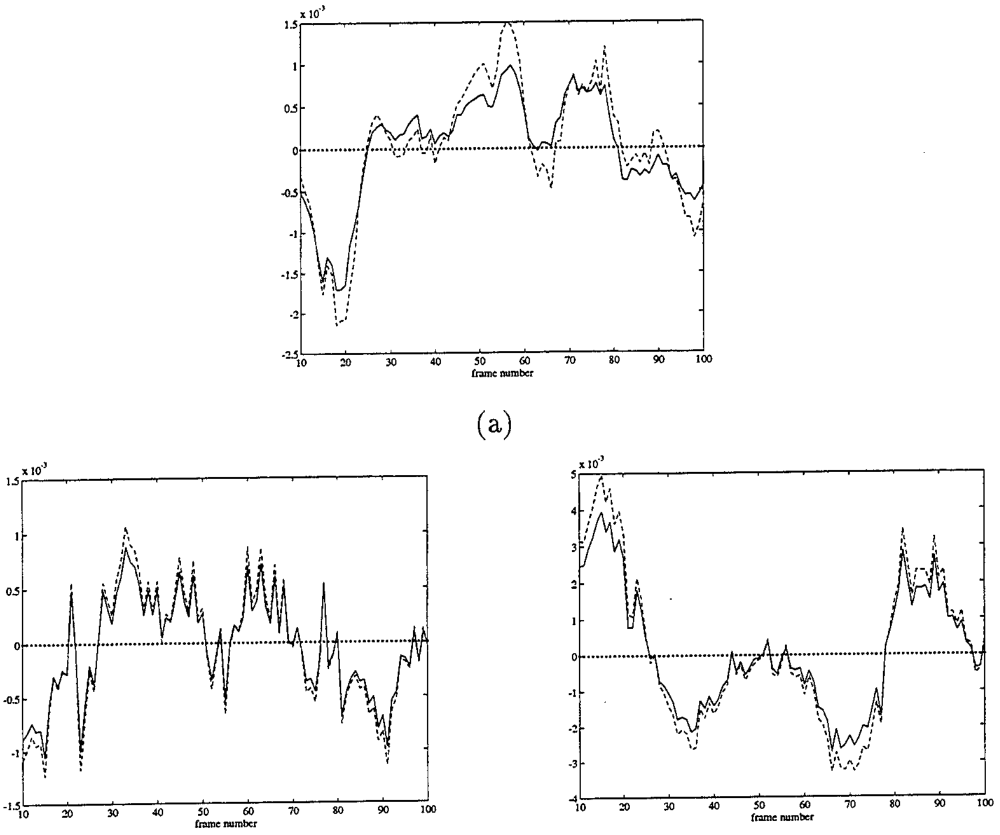
This chapter has presented a scheme for stabilizing a sequence acquired by a moving observer. In particular, the algorithm exploits the temporal information in the sequence and utilizes various visual cues. Image warpings derived from the 3D motion are used instead of other approximate 2D mappings, thereby capturing more closely the image motion resulting from the camera rotation.



(a) (b) (c) (d)

Figure 3.9: Sensitivities of **b** and **c** with respect to f_c in the horizontal direction (solid line) and x_c (dashed line) for Martin Marietta sequence: (a) b_1, (b) b_2, (c) c_1, (d) c_2.

The consideration of global stabilization, in addition, makes the algorithm more suitable to real applications. We analyze the nature of the resulting sequence. The use of kinetic laws for estimation purposes is carefully justified and tested in synthetic cases. Lastly, we study the removal of rotational motion from uncalibrated sequences, for which a scheme to alleviate the unstable estimation of the projectivity coefficients is proposed and tested. All results have been thoroughly tested and the resulting sequences have been made available to the interested reader.

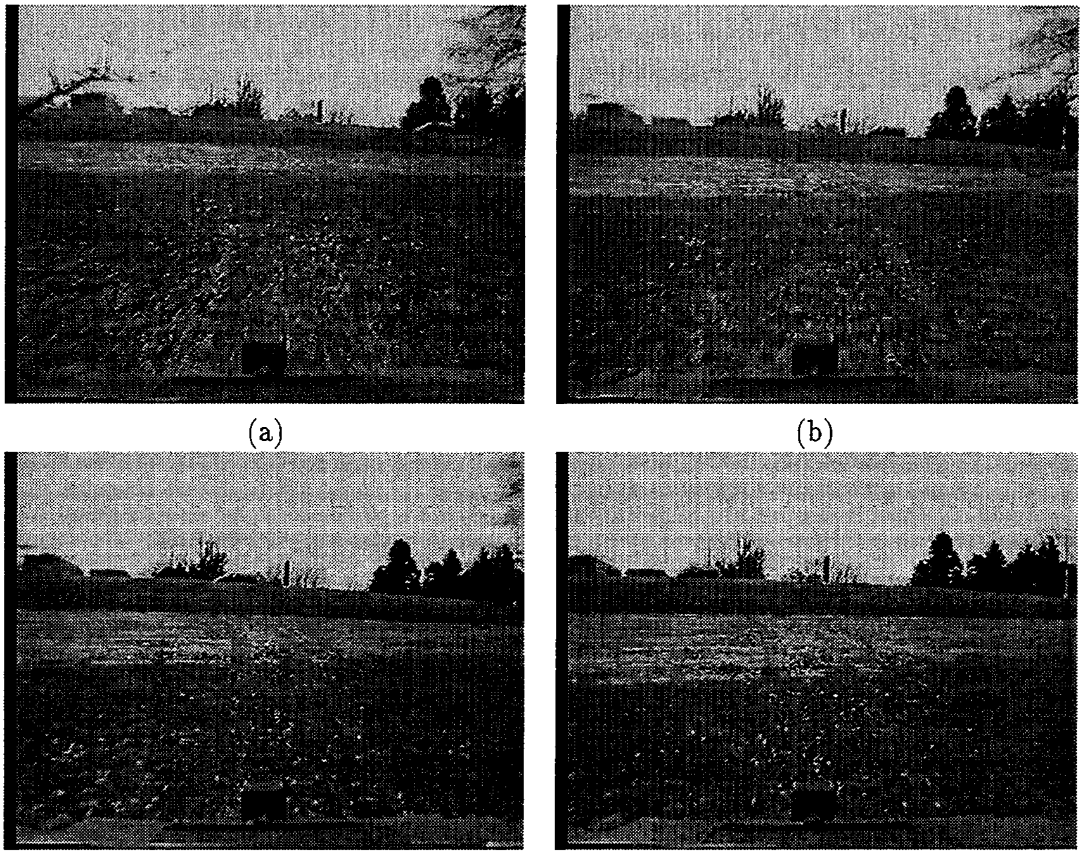


(a)

(b)

(c)

Figure 3.10: Angular velocities from the uncalibrated scheme (solid line) versus calibrated scheme (dashed line) for Martin Marietta sequence: (a) ω_x, (b) ω_y, (c) ω_z.



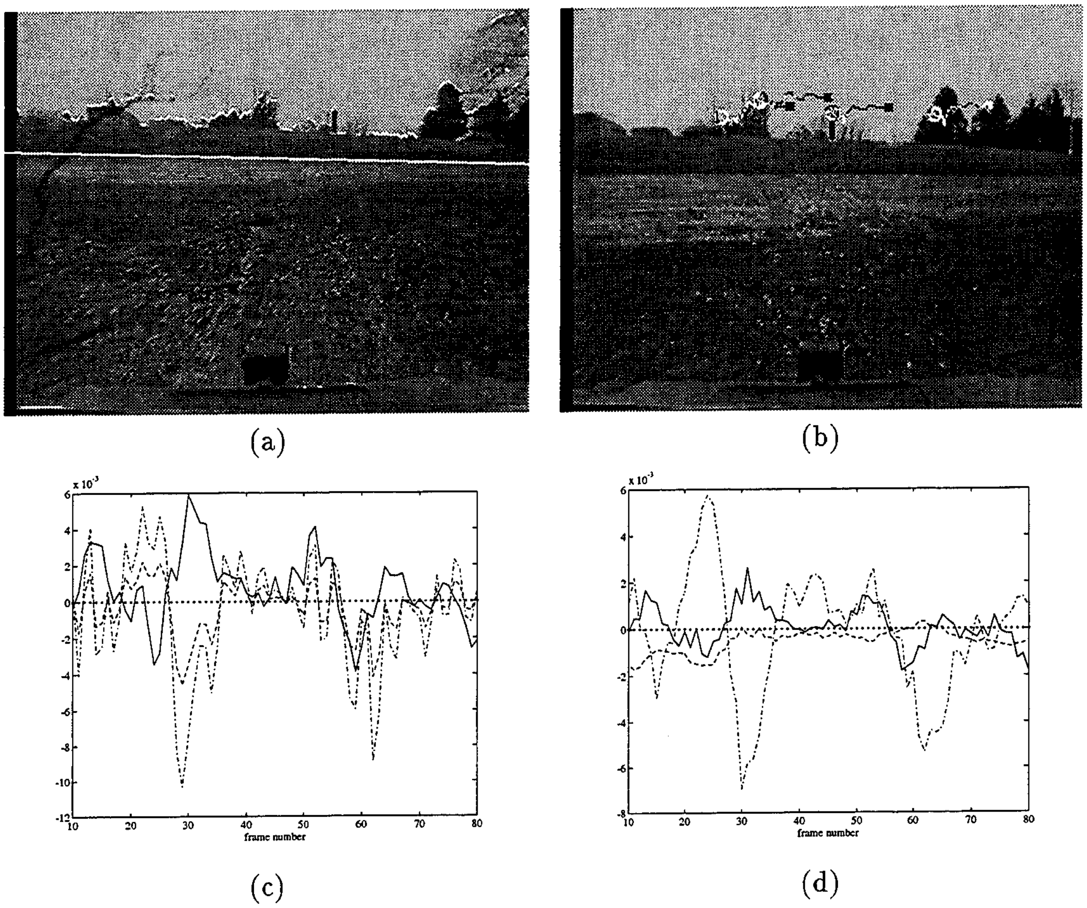
(a)

(b)

(c)

(d)

Figure 3.11: NIST sequence: (a) Image 10, (b) Image 35, (c) Image 60, (d) Image 80.

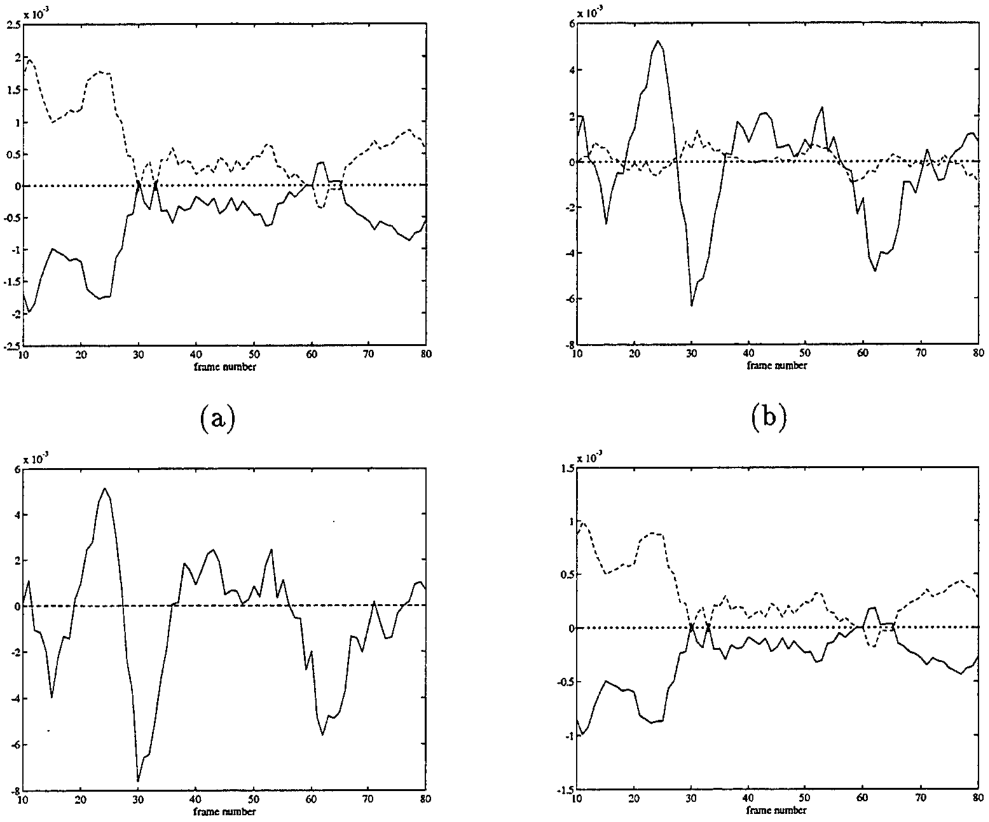

(a)

(b)

(c)

(d)

Figure 3.12: Experimental results from the calibrated scheme for the NIST sequence: (a) The detected line and partial object profiles, (b) Feature point trajectories, (c) LS estimates, (d) Fully stabilized estimates.



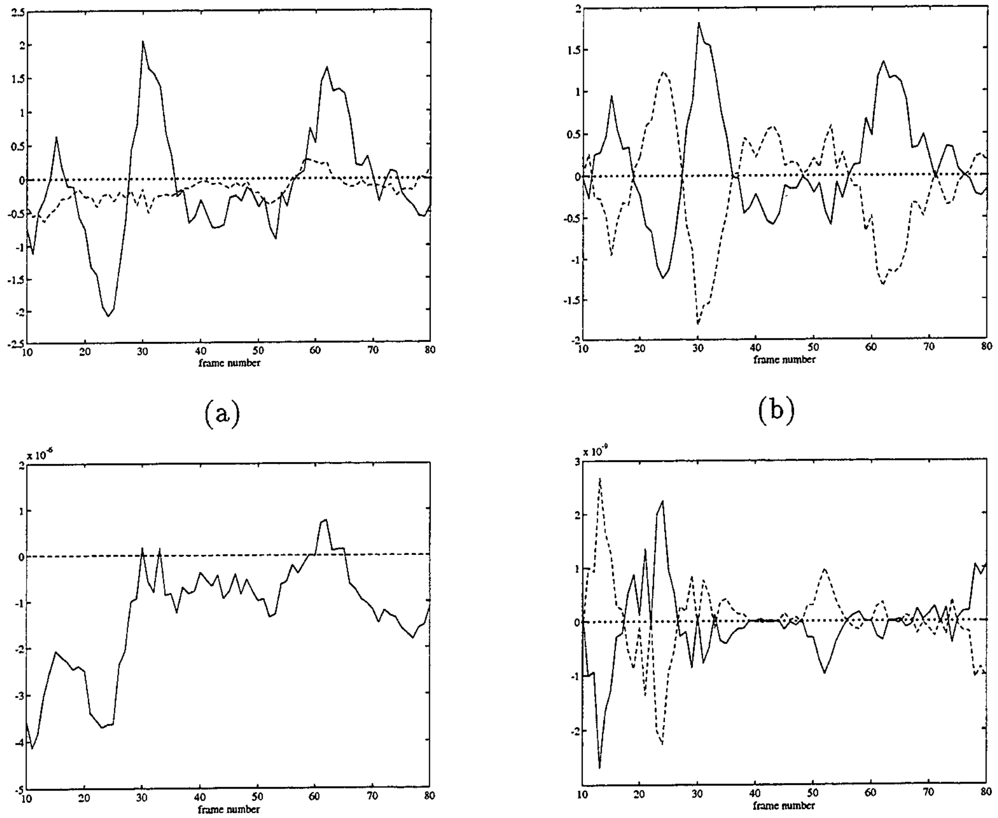
(a)

(b)

(c)

(d)

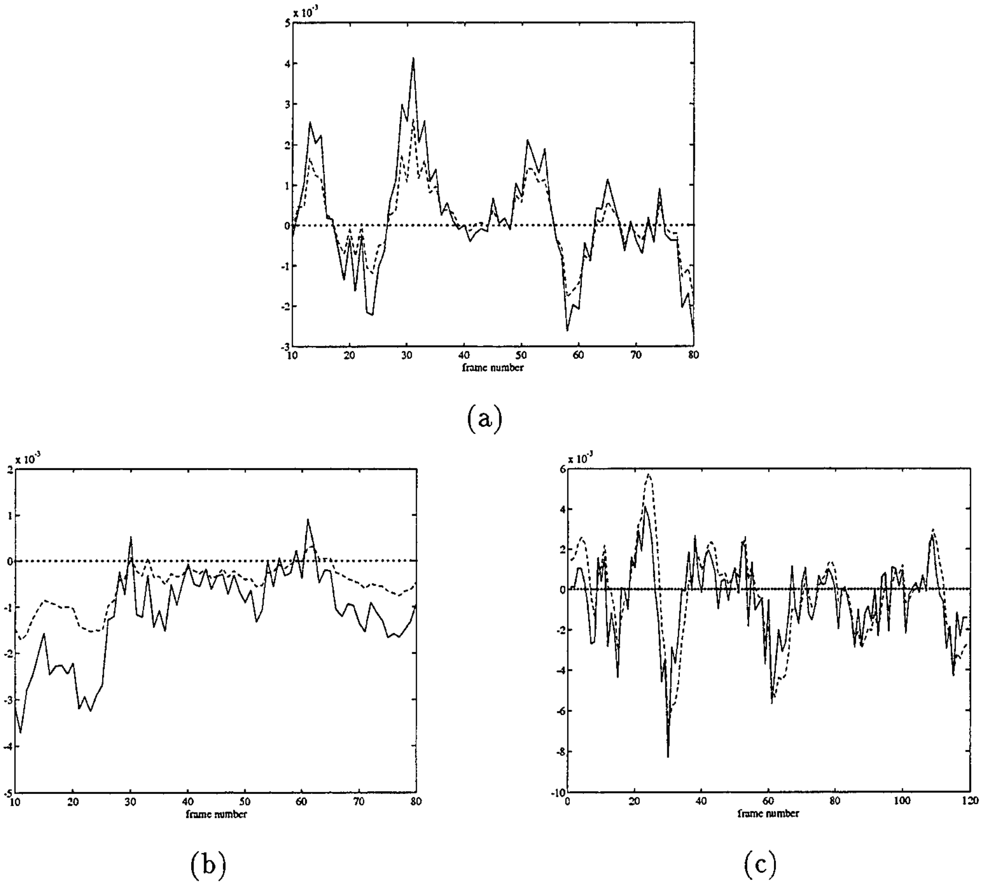
Figure 3.13: Sensitivities of **A** with respect to f_c in the horizontal direction (solid line) and x_c (dashed line) for NIST sequence: (a) a_{11}, (b) a_{12}, (c) a_{21}, (d) a_{22}.



(a) (b)

(c) (d)

Figure 3.14: Sensitivities of **b** and **c** with respect to f_c in the horizontal direction (solid line) and x_c (dashed line) for NIST sequence: (a) b_1, (b) b_2, (c) c_1, (d) c_2.



(a)

(b)

(c)

Figure 3.15: Angular velocities from the uncalibrated scheme (solid line) versus calibrated scheme (dashed line) for NIST sequence: (a) ω_x, (b) ω_y, (c) ω_z.

Chapter 4

Structure from Selective Stabilization

4.1 Introduction

We address the interpretation of visual motion from selective stabilization in this chapter. The consideration of oscillatory motion is important for many navigation related applications. For example, for an active vision system, the separation of residual oscillatory motion from smooth motion is useful for achieving fixation. Another example where selective stabilization is important is teleoperation in which the vehicle needs to be remotely controlled. An image sequence unperturbed by the residual oscillatory motion, while still preserving smooth motion, is highly desirable: the teleoperator needs to fully evaluate the effects of climbing and steering. Furthermore, while an approximate selective stabilization scheme could be attempted by simple low-pass filtering of the computed total rotational components, a scheme which more closely resembles mechanical stabilization is of more interest.

To achieve selective stabilization, we first consider the recovery of parameters relevant to total rotation which is the sum of smooth rotation and oscillatory rotation. This procedure has been described in the previous chapter. Subsequently, we seek out the separation of smooth rotation and residual oscillatory rotation. The essence is the exploitation of appropriate dynamic laws for both phenomena. By assuming the desired smooth rotation results in smooth attitude change of a moving vehicle, a kinematic law is employed to model the behavior of corresponding smooth rotation parameters. The residual oscillatory rotation is captured by a kinetic model. A four-wheel vehicle model described in the previous chapter is considered for this purpose. Moreover, it is observed that, on some occasions, the smooth rotation and the residual oscillatory rotation are not completely coupled. If the high frequency yaw motion is negligible, steering is directly characterized by total rotation. Similarly, if the banking behavior is not considered, the roll motion is also observable from the total rotation. Therefore, only the behavior of climbing and pitch motion is always coupled. However, the climbing phenomenon does not exist at all times. It only lasts during the short period whenever the vehicle moves between surfaces with different slopes, for example, from a horizontal surface to an uphill path. Consequently, we assume that only climbing and pitch phenomena need to be separated in our selective stabilization scheme. A detection scheme is then designed for the detection of start and end of climbing. Based on the detection results, the algorithm employs dynamic laws to achieve selective stabilization.

Subsequently, we estimate the relative positions of close scene points with respect to the vehicle. Much of the earlier work estimates these relative positions without consideration of the residual oscillatory rotation. Decisions or actions based on the resulting estimates are likely to be inappropriate. With selective stabilization achieved, we estimate the 3D locations of a set of close features in a more desirable frame of reference, which is less perturbed by the oscillatory rotation.

The organization of this chapter is as follows. Section 4.2 discusses the separability of smooth rotation and residual oscillatory rotation. The motion/structure recovery algorithm including the detection of the start and end of smooth rotation as well as estimation of structure parameters is presented in Section 4.3. Section 4.4 reports results of synthetic experiments. Conclusions are given in Section 4.5.

4.2 Selective Stabilization

The inherent nature of video signal processing implies that temporal information present in the sequence should be exploited. We show that suitable dynamic laws also facilitate selective stabilization in this section.

For convenience, denote the angular velocities arising from the smooth rotation by $\boldsymbol{\omega}_s$ and from the oscillatory rotation by $\boldsymbol{\omega}_{us}$ respectively. Consider $\boldsymbol{\omega}_s$ first. Since the smooth rotation is due to steering and climbing, it is assumed that the resulting attitude changes smoothly over time. A simple kinematic law is therefore employed to describe the behavior of $\boldsymbol{\omega}_s : (\omega_{sx}, \omega_{sy}, \omega_{sz})^T$,

$$\dot{\boldsymbol{\omega}}_s = \mathbf{0} \tag{4.1}$$

On the other hand, recall the four-wheel vehicle model described in the previous chapter. The residual oscillatory rotation $\mathbf{x}_{us}$ is described by (3.34), i.e.

$$\dot{\mathbf{x}}_{us} = \mathbf{\Phi}_{us}\mathbf{x}_{us} + \mathbf{\Gamma}_{us}\mathbf{u}_{us} \tag{4.2}$$

Instead of using sensors such as accelerometers to obtain $\mathbf{u}_{us}$, we employ an approximate law to describe $\mathbf{x}_{us}$ as

$$\dot{\mathbf{x}}_{us} = \mathbf{\Phi}_{us}\mathbf{x}_{us} \tag{4.3}$$

It is shown in Appendix D that this approximation introduces an additional error in separating smooth rotation and residual oscillatory rotation. However, the error remains bounded.

Next, define the stabilized coordinate system $\mathcal{V}_s(t)$ whose orientation is determined by the smooth rotation, and whose center is located at the center of gravity of the vehicle. Consider also the unstabilized coordinate system $\mathcal{V}(t)$, whose attitude is in addition affected by the residual oscillatory rotation, and with its origin shifted from the center of gravity due to bounce. Then, from two images acquired at times $t - \Delta t$ and t, we observe the total angular velocity accounting for the net attitude change between $\mathcal{V}(t-\Delta t)$ and $\mathcal{V}(t)$. In other words, if $\mathbf{R}(\omega; t, t-\Delta t)$ aligns the coordinate system $\mathcal{V}(t-\Delta t)$ with $\mathcal{V}(t)$, $\mathbf{R}_s(\boldsymbol{\omega}_s; t, t-\Delta t)$ takes into account the attitude difference between $\mathcal{V}_s(t-\Delta t)$ and $\mathcal{V}_s(t)$, and $\mathbf{R}_{us}(\theta, \phi; t)$ accounts for the additional attitude due to the oscillatory rotation at time t,

$$\mathbf{R}_{us}(\theta, \phi; t) = \begin{bmatrix} 1 & \phi(t) & -\theta(t) \\ -\phi(t) & 1 & 0 \\ \theta(t) & 0 & 1 \end{bmatrix} \tag{4.4}$$

then it can be shown that

$$\mathbf{R}(\omega; t, t-\Delta t) = \mathbf{R}_{us}(\theta, \phi; t)\mathbf{R}_s(\boldsymbol{\omega}_s; t, t-\Delta t)\mathbf{R}_{us}^{-1}(\theta, \phi; t-\Delta t) \tag{4.5}$$

Subsequently, assume that $\mathbf{R}$ and $\mathbf{R}_s$ can be approximated as

$$\mathbf{R}(\omega; t, t-\Delta t) \approx \mathbf{I} - \boldsymbol{\Omega}\Delta t \tag{4.6}$$

$$\mathbf{R}_s(\boldsymbol{\omega}_s; t, t-\Delta t) \approx \mathbf{I} - \boldsymbol{\Omega}_s\Delta t \tag{4.7}$$

with

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\Omega}_s = \begin{bmatrix} 0 & -\omega_{sz} & \omega_{sy} \\ \omega_{sz} & 0 & -\omega_{sx} \\ -\omega_{sy} & \omega_{sx} & 0 \end{bmatrix} \tag{4.8}$$

Also,

$$\begin{aligned} \tfrac{\theta(t)-\theta(t-\Delta t)}{\Delta t} &\approx \dot{\theta}(t) \\ \tfrac{\phi(t)-\phi(t-\Delta t)}{\Delta t} &\approx \dot{\phi}(t) \end{aligned} \tag{4.9}$$

Then the total angular velocity $\boldsymbol{\omega}$ is related to the angular velocity corresponding to the smooth rotation and residual oscillatory rotation by

$$\boldsymbol{\omega} = \boldsymbol{\omega}_s + \boldsymbol{\omega}_{us} \tag{4.10}$$

with

$$\boldsymbol{\omega}_{us} \equiv (0, \dot{\theta}, \dot{\phi})^T \tag{4.11}$$

Since the angular velocity component of the yaw motion is assumed to be zero, we only focus on the separability of the other two phenomena, i.e. climbing and pitch as well as banking and roll. Using the dynamic laws in (4.1) and (4.3), the following linear system can be constructed:

$$\left\{ \begin{array}{lcl} \dot{\mathbf{x}}_1 & = & \mathbf{G}\mathbf{x}_1 \\ \boldsymbol{\omega}_1 & = & \mathbf{F}\mathbf{x}_1 \end{array} \right. \tag{4.12}$$

where $\mathbf{x}_1 = (\omega_{sy}, \omega_{sz}, \mathbf{x}_{us}^T)^T$, $\boldsymbol{\omega}_1 = (\omega_y, \omega_z)^T$,

$$\mathbf{G} = \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 4} \\ \mathbf{0}_{4\times 2} & \boldsymbol{\Phi}_{us} \end{bmatrix} \tag{4.13}$$

with $\mathbf{0}_{2\times2}, \mathbf{0}_{2\times4}$ and $\mathbf{0}_{4\times2}$ being zero matrices of conformable dimensions,

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.14}$$

The separability of smooth rotation and residual oscillatory rotation can now be verified by checking the rank of the observability matrix $\mathbf{O}$,

$$\mathbf{O} = \begin{bmatrix} \mathbf{F} \\ \mathbf{FG} \\ \vdots \\ \mathbf{FG}^5 \end{bmatrix} \tag{4.15}$$

Except in some special cases in which the matrix $\boldsymbol{\Phi}_{us}$ is singular, $\mathbf{O}$ in general has full rank. Therefore, by employing different dynamic laws to describe the smooth rotation and residual oscillatory rotation, it is possible to separate the two rotation phenomena.

The next section presents schemes for the detection of the start and end of smooth rotation, and the estimation of 3D locations of close feature points in the stabilized coordinate system.

4.3 Parameter Estimation

The incorporation of appropriate dynamic laws to describe smooth rotation and residual oscillatory rotation has made selective stabilization possible. Since smooth rotation does not last for a long time, the detection of the beginning or end of smooth rotation, especially climbing, facilitates the task of selective stabilization. In addition to the total rotational parameter estimation scheme mentioned in Section 3.5, this section describes the other schemes which comprise our algorithm for off-road vehicular navigation.

4.3.1 Maneuver Detection / Selective Stabilization

As discussed earlier, when the high frequency yaw motion is negligible, steering is directly known from the total angular velocity. Similarly, the phenomenon of banking only lasts for short periods and more importantly, it does not have later effects on the direction in which the vehicle travels. Banking is therefore also neglected, and consequently, the angular velocity component explaining roll motion is obtained directly. However, climbing is not negligible; the vehicle generally moves along a direction resulting from climbing. Thus, the separation of climbing and pitch motion is important. We first describe a detection scheme designed to detect the beginning and end of the climbing phenomenon.

Recall the image motion expressed in (3.4). When a distant point is considered, its vertical movement is described by

$$\dot{x} = \omega_x[(x - x_c)(y - y_c)] - \omega_y[f_c + (x - x_c)^2] + \omega_z(y - y_c) \tag{4.16}$$

If the corresponding projection point is close to the vertical axis, i.e. $y - y_c \approx 0$, then

$$\dot{x} \approx -\omega_y[f_c + (x - x_c)^2] \tag{4.17}$$

In other words, the vertical movements of such features are dominantly affected by climbing and pitch motions. Since a zero order dynamic law is assumed for the total angular velocity, it is expected that when the vehicle performs maneuvers (i.e. starts climbing or finishes climbing), large errors occur at these instants in predicting the vertical movements of corresponding points. We define $[\mathbf{z}_m]_x$ to be the vector consisting of the vertical coordinates of N such feature points, $\mathbf{p}_{m1}, \ldots, \mathbf{p}_{mN}$, i.e.

$$[\mathbf{z}_m]_x = ([\mathbf{p}_{m1}]_x, \ldots, [\mathbf{p}_{mN}]_x)^T \tag{4.18}$$

and the test statistic δ at time t as

$$\delta(t) \equiv ([\mathbf{z}_m(t)]_x - [\hat{\mathbf{z}}_m(t)]_x)^T \Sigma_m^{-1}(t)([\mathbf{z}_m(t)]_x - [\hat{\mathbf{z}}_m(t)]_x) \tag{4.19}$$

where $\hat{\mathbf{z}}_m(t)$ denotes the predicted locations of these N feature points at time t; its j^{th} element is computed by

$$\hat{\mathbf{p}}_m(t) = [\mathbf{c}^T\mathbf{p}_m(t - \Delta t) + 1]^{-1}[\mathbf{A}\mathbf{p}_m(t - \Delta t) + \mathbf{b}] \tag{4.20}$$

where the predicted angular velocity is used in calculating $\mathbf{A}, \mathbf{b}, \mathbf{c}$. $\Sigma_m(t)$, on the other hand, is the covariance matrix of the innovation vector $[\mathbf{z}_m(t)]_x - [\hat{\mathbf{z}}_m(t)]_x$; it can easily be computed in our EKF formulation. Then, based on $\delta(t)$, we declare that the vehicle undergoes maneuvers at $t - \Delta t$ when

$$\delta(t) > TH \tag{4.21}$$

with TH a threshold value. The types of maneuvers corresponding to the beginning or end of climbing are then determined. If the vehicle starts climbing at a rate $\omega_{sy}(t)$ much larger than the pitching velocity $\dot{\theta}(t)$, the resulting maneuver can be identified by comparing $\delta(t)$ to previous non-maneuver test statistics $\delta(t-\Delta t), \delta(t-2\Delta t), \ldots, \delta(t-m\Delta t)$, with m the size of a moving window. Similarly, if the type of maneuver leads to the end of the climbing phenomenon, the scheme is able to react instantaneously.

There are situations in which the $\omega_{sy}(t)$, at maneuver instants, is less than $\dot{\theta}(t)$. If we regard $\omega_{sy}(t)$ as the desired signal, and $\dot{\theta}(t)$ as the additive noise, we are dealing with a problem involving a small signal to noise ratio. The test statistic $\delta(t)$ may become inappropriate since the maneuver test statistic does not manifest itself in larger values. We therefore impose another conservative criterion. This criterion cannot detect maneuvers instantaneously. However, it notices the event a few frames later and prevents the divergence of the whole algorithm.

Specifically, observe that when the vehicle only undergoes pitch motion, $\mathbf{p}_{m1}, \ldots, \mathbf{p}_{mN}$ are likely to oscillate around their own reference positions. The conservative detection criterion monitors the vertical movements of these points with some frequency less than the sequence acquisition rate. When these N points start to move away from the reference positions, the algorithm reports the start of climbing. Reprocessing a small set of images is then performed. Afterwards, when these points oscillate around new reference positions, the climbing is assumed to end.

We now proceed to address the separation of climbing and pitch motion. First, let the vector $\mathbf{y}$ consist of the quantities of interest. When the start of climbing is detected,

$$\mathbf{y} = (\omega_{sy}, \theta, \dot{\theta})^T; \tag{4.22}$$

and otherwise,

$$\mathbf{y} = (\theta, \dot{\theta})^T \tag{4.23}$$

Next, we use a Kalman Filter (KF) to achieve selective stabilization. The plant equation is obtained by employing the dynamic laws described in (4.1) and (4.3). The observation equation can be written as

$$\omega_y = \mathbf{H}\mathbf{y} \tag{4.24}$$

where when ω_{sy} is included,

$$\mathbf{H} = [1 \quad 0 \quad 1] \tag{4.25}$$

and otherwise,

$$\mathbf{H} = [0 \quad 1] \tag{4.26}$$

Using the assumptions that

$$\begin{cases} \omega_{sx} & = & \omega_x \\ \dot{\phi} & = & \omega_z \end{cases} \tag{4.27}$$

the smooth rotation and the residual oscillatory rotation are separated. The next section discusses the use of selective stabilization to recover the 3D locations of feature points which are close to the vehicle.

4.3.2 Structure from Selective Stabilization

Recall that the motion of a close 3D point relative to the stabilized coordinate system $\mathcal{V}_s$ is described by

$$\dot{\mathbf{P}}_{\mathcal{V}_s} = -\boldsymbol{\omega}_s \times \mathbf{P}_{\mathcal{V}_s} - V_s \tag{4.28}$$

with $\mathbf{P}_{\mathcal{V}_s}$ denoting the coordinates of the point in $\mathcal{V}_s$, $\boldsymbol{\omega}_s$ being the angular velocity due to the smooth rotation, and $\mathbf{V}_s$ being the vehicle's translational velocity under no disturbances. When the vehicle undergoes constant smooth rotation and translation between $t - \Delta t$ and t, we have

$$\mathbf{P}_{\mathcal{V}_s}(t) = \mathbf{R}_s(\boldsymbol{\omega}_s; t, t - \Delta t)\mathbf{P}_{\mathcal{V}_s}(t - \Delta t) + \mathbf{T}_s(\boldsymbol{\omega}_s; t, t - \Delta t) \tag{4.29}$$

where $\mathbf{R}_s$ accounts for the resulting attitude change as explained in (4.7), while $\mathbf{T}_s$ can be shown [60] to be

$$\mathbf{T}_s(\boldsymbol{\omega}_s; t, t - \Delta t) = -\left[\mathbf{I}\Delta t - \frac{1 - \cos(|\boldsymbol{\omega}_s|\Delta t)}{|\boldsymbol{\omega}_s|^2}\boldsymbol{\Omega}_s + \frac{|\boldsymbol{\omega}_s|\Delta t - \sin(|\boldsymbol{\omega}_s|\Delta t)}{|\boldsymbol{\omega}_s|^3}\boldsymbol{\Omega}_s^2\right]\mathbf{V}_s \tag{4.30}$$

with $\boldsymbol{\Omega}_s$ being defined in (4.8) and $|\boldsymbol{\omega}_s|$ denoting the magnitude of $\boldsymbol{\omega}_s$. In particular, observing that the translational movement of a vehicle is along its longitudinal axis most of the time [33], $\mathbf{V}_s$ is simplified to

$$\mathbf{V}_s = (0, 0, v_s)^T \tag{4.31}$$

with v_s being the forward speed. Since v_s can easily be obtained from the odometer, we assume that it is known.

The relative motion of the point with respect to the unstabilized coordinate system $\mathcal{V}$ is then obtained as

$$\mathbf{P}_{\mathcal{V}}(t) = \mathbf{R}_{us}(\theta, \phi; t)\left\{\mathbf{P}_{\mathcal{V}_s}(t) - \mathbf{T}_{us}(t)\right\} \tag{4.32}$$

where $\mathbf{T}_{us}$ is due to the bounce of the vehicle,

$$\mathbf{T}_{us}(t) = (x_c(t), 0, 0)^T \tag{4.33}$$

The projection of the 3D point on the image is then computed by applying the perspective projection operator $\mathcal{P}$,

$$\mathbf{p} = \mathcal{P}(\mathbf{P}_{\mathcal{V}} - \mathbf{d}) + \mathbf{p}_c \tag{4.34}$$

where $\mathbf{d}$ specifies the position of the camera with respect to $\mathcal{V}$; it remains constant because the camera is mounted rigidly on the vehicle. Since the effect of $\mathbf{T}_{us}$ is usually too small to be observed from the images, it is not considered. Therefore, by tracking the points of interest over the sequence as well as using the estimates from selective stabilization, we employ another EKF to estimate $\mathbf{P}_{\mathcal{V}_s}$. The corresponding plant equations for M_s such points are

$$\mathbf{P}_{\mathcal{V}_s,j}(t_{i+1}) = \mathbf{R}_s(\boldsymbol{\omega}_s; t_{i+1}, t_i)\mathbf{P}_{\mathcal{V}_s,j}(t_i) + \mathbf{T}_s(\boldsymbol{\omega}_s; t_{i+1}, t_i) \quad j = 1 \cdots M_s \tag{4.35}$$

The measurement equations, on the other hand, are

$$\mathbf{p}_j = \mathcal{P}(\mathbf{P}_{\mathcal{V},j} - \mathbf{d}) + \mathbf{p}_c \quad j = 1 \cdots M_s \tag{4.36}$$

The undesired oscillatory phenomena, when estimated with respect to the unstabilized coordinate system $\mathcal{V}$, are removed.

This concludes our scheme for off-road navigation using selective stabilization. The next section presents simulation results.

4.4 Synthetic Experiments

In this section, we illustrate the proposed approach through synthetic experiments. Two experiments were performed. One corresponds to the case of high signal to noise ratio in which the attitude change due to climbing is **much** larger than the change arising from pitch motion. The other one deals with the case involving low signal to noise ratio.

Consider the scenario in which the vehicle undergoes steering for the first one second, with a rate of $\omega_{sx} = 0.1$ rad $\cdot$ s^{-1}. Immediately after the steering, the vehicle starts climbing over a hill. The smooth rotation due to climbing exists for two seconds. The climbing rates are set to $\omega_{sy} = 0.1$ rad$\cdot$s^{-1} and 0.01 rad$\cdot$s^{-1}, in the high and low signal to noise ratio situations respectively. Finally, the vehicle moves uphill along a straight path for the last two seconds, i.e. no smooth rotation occurs. The forward speed of the vehicle is maintained at $v_s = 10.0\text{m} \cdot \text{s}^{-1}$.

The surface irregularities, on the other hand, are modeled using first order Markov processes; their coefficients depend on the terrain roughness and the vehicle's forward speed. The excitations to the vehicle are given by

$$\dot{x}_{0i}(t) = -avx_{0i}(t) + n_{av}(t) \quad i = 1 \cdots 4 \tag{4.37}$$

where n_{av} is the white, Gaussian driving noise with mean zero and variance $2\sigma^2 av$. The values of a and σ vary for different surfaces. We choose $\sigma = 0.012$m and $a = 0.8\text{m}^{-1}$ [37]. The nominal values of the vehicle's parameters are listed in Table 4.1. According to the equations of motion derived in Appendix A, we synthesize the oscillatory behavior of the vehicle and illustrate it in Figure 4.1 (only pitch and roll phenomena are shown). Together with the smooth motion, the behavior of the vehicle is obtained.

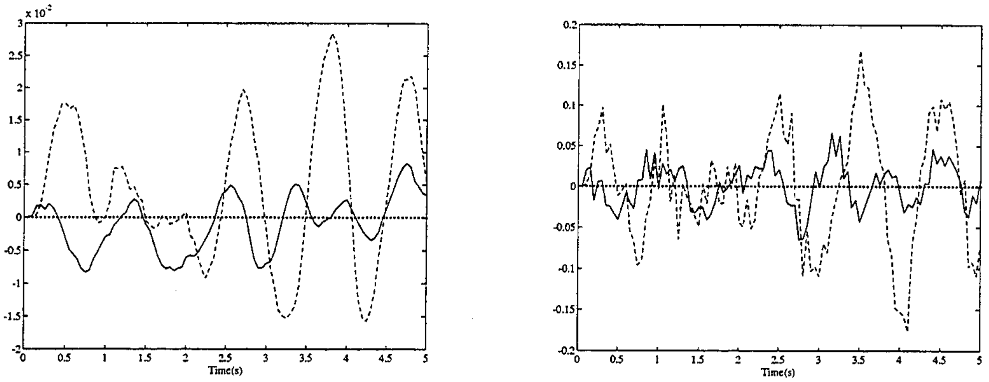
Table 4.1: The nominal parameter values of the four-wheel vehicle model.

M_B	I_{yy}	I_{zz}	M_{wf}
1710.0kg	1031.3kg $\cdot$ m^2	201.8kg $\cdot$ m^2	57.5kg
M_{wr}	K_T	C_f (C_r)	K_f
75.0kg	200.0 kN $\cdot$ m^{-1}	1.0 kN $\cdot$ m^{-1} $\cdot$ s^{-1}	18.0 kN $\cdot$ m^{-1}
K_r	$T_{s,r}$ $(T_{s,l})$	W_A	W_B
10.0 kN $\cdot$ m^{-1}	0.6m	1.4m	1.3m

In addition, during driving, a sequence of images is acquired at $20Hz$ from an on-board camera (each image has size 2.0 × 2.0 and resolution 2000 × 2000), while a set of features including eight **distant** points and four **close** points are tracked. The coordinates of these points with respect to the initial frame of reference are listed in Table 4.2.

Table 4.2: The initial 3D locations of the feature points: the first eight points correspond to distant features, while the last four are considered to be close features.

Point	3D coordinates			Point	3D coordinates			Point	3D coordinates		
1	1260	36	1800	5	1620	1440	1800	9	45.5	-45.5	65.0
2	1600	-20	2000	6	1400	-1400	2000	10	46.4	-23.2	58.0
3	-1840	-69	2300	7	1150	-920	2300	11	45.0	30.0	60.0
4	-1575	70	1750	8	875	875	1750	12	42.0	56.0	70.0



(a) (b)

Figure 4.1: An example of the pitch (solid line) and roll (dashed line) behavior of the nominal vehicle. (a) $\theta(t)$ and $\phi(t)$, (b) $\dot{\theta}(t)$ and $\dot{\phi}(t)$.

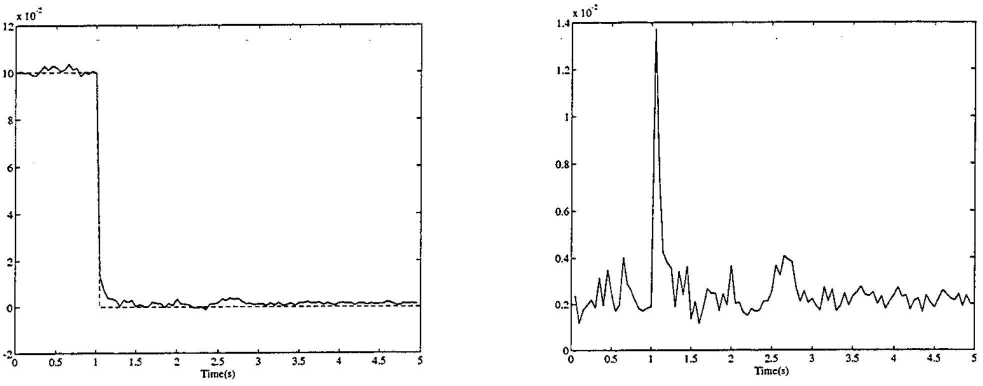
The algorithm is now applied to estimate the total angular velocity, to separate the rotation due to climbing and pitch motion, as well as to obtain the 3D locations of four close features with respect to the stabilized coordinate system.

4.4.1 High Signal to Noise Ratio

We consider here the situation in which the vehicle climbs uphill with a high climbing rate, $\omega_{sy} = 0.1 \text{ rad} \cdot \text{s}^{-1}$. The performance of the algorithm is evaluated from 50 Monte Carlo trials. Using the image stabilization technique described in the previous chapter, the total angular velocity is first computed from eight listed distant points. The resulting x and z components are directly regarded as the steering rate ω_{sx} and the roll velocity $\dot{\phi}$. Figures 4.2 and 4.4 respectively show the corresponding bias and root mean-square errors of these estimates. The estimate of ω_y is illustrated in Figure 4.3a. The test statistic δ, defined in (4.19), is also calculated using the vertical movements of points 1–4, and is shown in Figure 4.3b. As seen from Figure 4.3b, δ exhibits large values when maneuvers occur ($t = 1.0$ s and $t = 3.0$ s). Thus the start and end of climbing can be detected instantaneously. Consequently, we take ω_{sy} into consideration from $t = 1.0$ s to $t = 3.0$ s. The results of selective stabilization are shown in Figure 4.5. It can be seen that the algorithm separates both climbing and pitch phenomena quite well. After selective stabilization has been achieved, these estimates are used to obtain 3D positions of close features, i.e. points 9–12. The initial guesses about these points are listed in Table 4.3, while the estimates are shown in Figure 4.6. The respective estimates exhibit similar behavior; therefore only the results for the 12th point are displayed. It is observed that, when the signal to noise ratio is high, the algorithm provides satisfactory estimates which should be useful for off-road vehicular navigation.

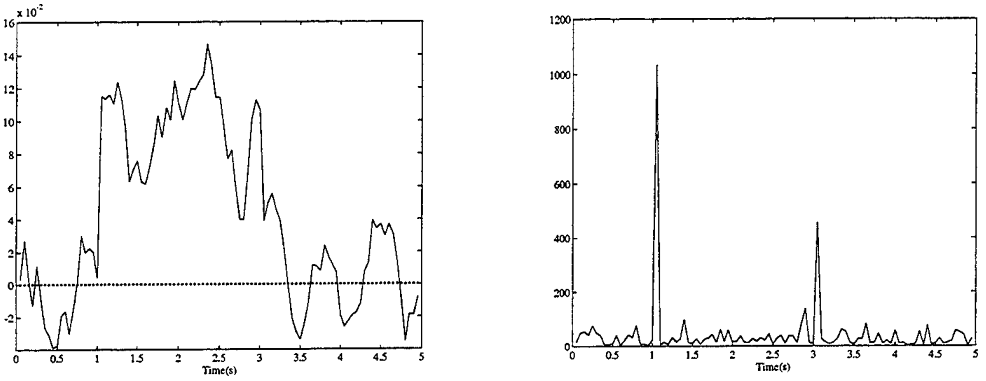
4.4.2 Low Signal to Noise Ratio

We now present some results corresponding to the low signal to noise ratio case, i.e. $\omega_{sy} = 0.01 \text{ rad} \cdot \text{s}^{-1}$. This corresponds to the scenario that the vehicle is moving uphill where the slope of the hill is smaller. Again, the performance is evaluated from 50 Monte Carlo trials. Figures 4.7a and 4.7b show the estimates of the total angular velocity ω and the test statistic δ. Since the x and z components of ω exhibit similar behavior to that in the high signal to noise ratio case, they are not



(a) (b)

Figure 4.2: Estimates of the steering rate ω_{sx}, in the high signal to noise ratio case, from 50 Monte Carlo trials. (a) True (dashed line) and estimated (solid line), (b) root mean-square error.



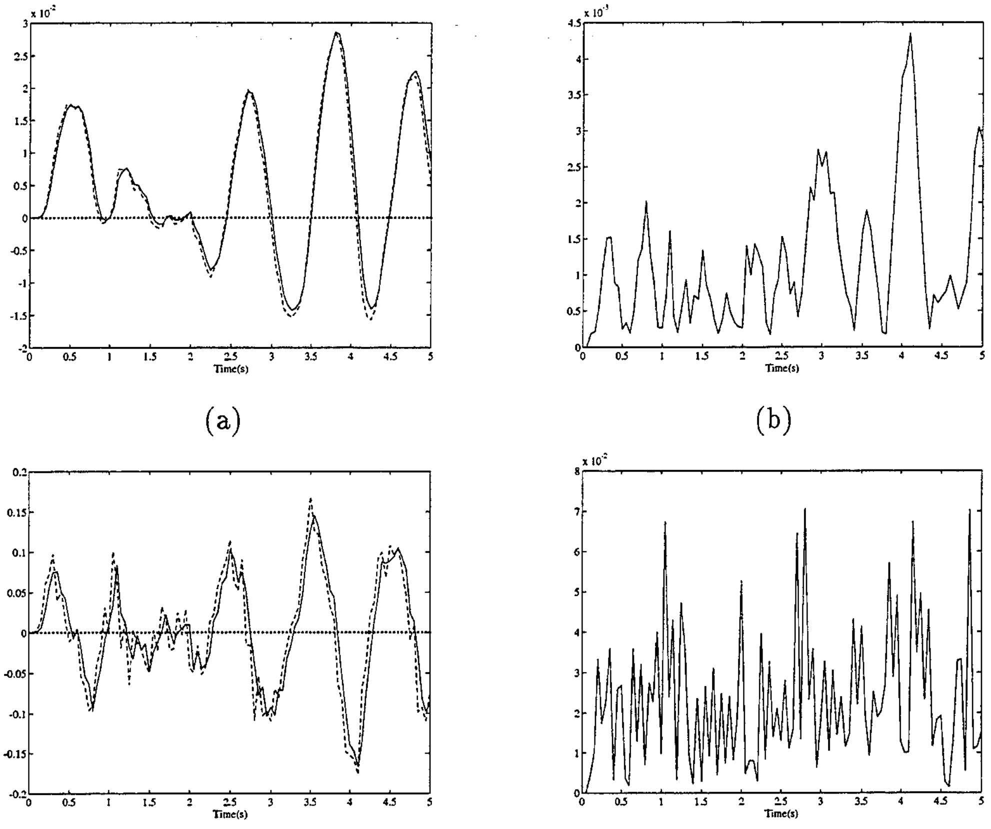
(a) (b)

Figure 4.3: Estimate of the climbing-pitch rate ω_y and test statistic δ, in the high signal to noise ratio case, from 50 Monte Carlo trials. (a) ω_y, (b) δ.

displayed here. In this low signal to noise ratio case, contrast with Figure 4.3b, the test statistic δ does not manifest large values at $t = 1.0$ s and $t = 3.0$ s. The start and end of climbing are hard to detect instantaneously. However, by monitoring the vertical movements of the first four distant points at the rate of $2Hz$ (the original sequence is acquired at $20Hz$), as shown in Figure 4.8, the scheme observes that these points move away from their first reference positions at $t = 2.5$ s. The algorithm then reprocesses the sequence from the image acquired at $t = 0.5$ s. Similarly, after $t = 3.5$ s, these points oscillate around new reference positions. The scheme declares that the climbing ends at $t = 3.5$ s. Selective stabilization results based on these conservative decisions are shown in Figure 4.10. As seen from this figure, the estimate of ω_{sy} is noisier in this case. This

Table 4.3: Initial guesses of 3D locations of points 9–12.

Point	3D coordinates		
9	42.3	-42.9	60.0
10	40.4	-20.5	50.0
11	37.9	25.6	50.0
12	36.2	48.9	60.0



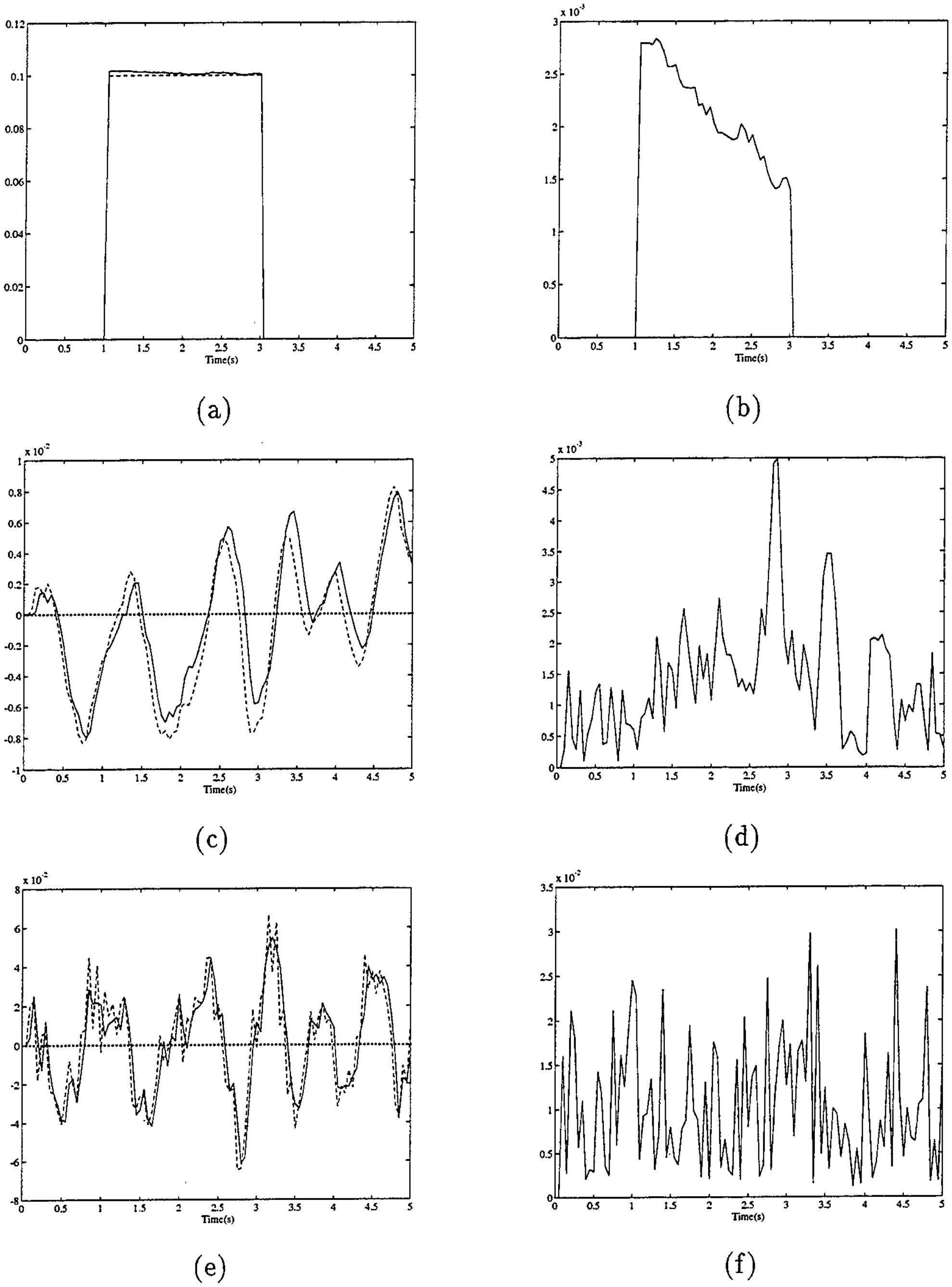
(c) (d)

Figure 4.4: Estimates of roll-related parameters, in the high signal to noise ratio case, from 50 Monte Carlo trials. (a) ϕ: true (dashed line) and estimated (solid line) (b) ϕ: root mean-square error (c) $\dot{\phi}$: true (dashed line) and estimated (solid line), (d) $\dot{\phi}$: root mean-square error.

leads to degradation in the estimates of θ and $\dot{\theta}$. The results corresponding to the 3D locations of close feature points are shown in Figure 4.9. Compared to Figure 4.6, these estimates exhibit a slightly larger bias. This is expected since the orientation of the stabilized coordinate system $\mathcal{V}_s$ is biased because of the smooth rotation is mistakenly introduced between $t = 3.0$ s and $t = 3.5$ s. Irrespective of these relatively larger biases, the corresponding absolute errors remain small. The structure information is therefore more or less preserved.

4.5 Conclusions

This chapter has presented a scheme for estimating the motion of a mobile robot as well as the relative positions of a set of discrete features. The algorithm employs different dynamic laws to capture different phenomena such as smooth rotation as well as pitch and roll motion. Various cues are first utilized for the estimation of total rotation. The use of an approximate kinetic law in achieving selective stabilization is then illustrated. Schemes for the detection of the beginning and end of smooth rotation are investigated. Due to the separation of the smooth rotation and the residual oscillatory rotation, the 3D locations of close features are easily estimated with respect to a less perturbed frame of reference. Finally, in addition to the applications in image understanding, results from the proposed selective stabilization scheme provide useful information for designing an active suspension system which compensates for the vibration more efficiently.

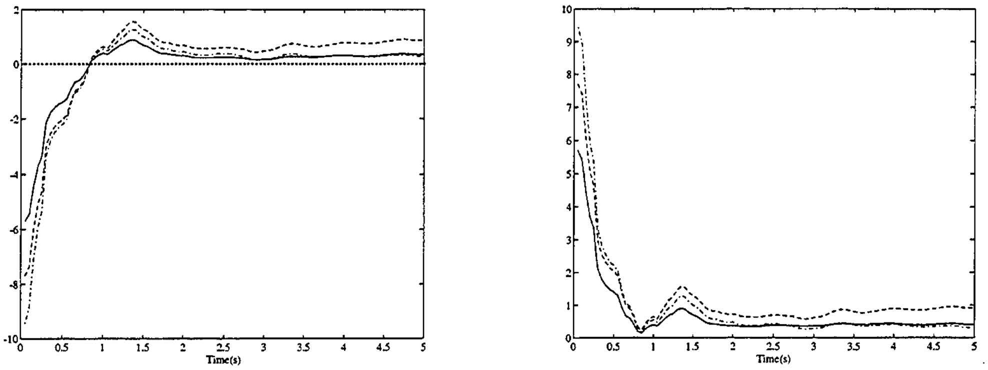


(a) (b)

(c) (d)

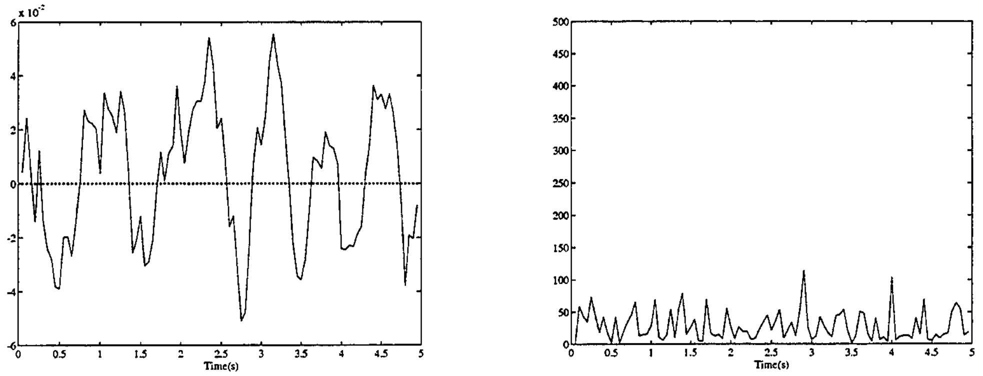
(e) (f)

Figure 4.5: Estimates of climbing-pitch parameters, in the high signal to noise ratio case, from 50 Monte Carlo trials. (a) ω_{sy}: true (dashed line) and estimated (solid line) (b) ω_{sy}: root mean-square error, (c) θ: true (dashed line) and estimated (solid line) (d) θ: root mean-square error, (e) $\dot{\theta}$: true (dashed line) and estimated (solid line), (f) $\dot{\theta}$: root mean-square error.



(a) (b)

Figure 4.6: Bias and root mean-square error in the estimates of 3D locations of close features, in the high signal to noise ratio case, from 50 Monte Carlo trials (solid line corresponds to the x coordinate, dashed line to the y coordinate, and dashed-dot line to the z coordinate). (a) Bias in Point 12, (b) root mean-square error in Point 12.



(a) (b)

Figure 4.7: Estimate of the climbing-pitch rate ω_y and test statistic δ, in the low signal to noise ratio case, from 50 Monte Carlo trials. (a) ω_y, (b) δ.

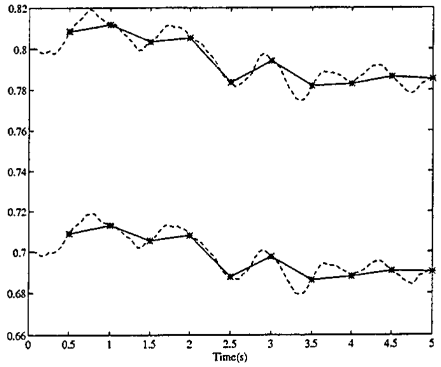
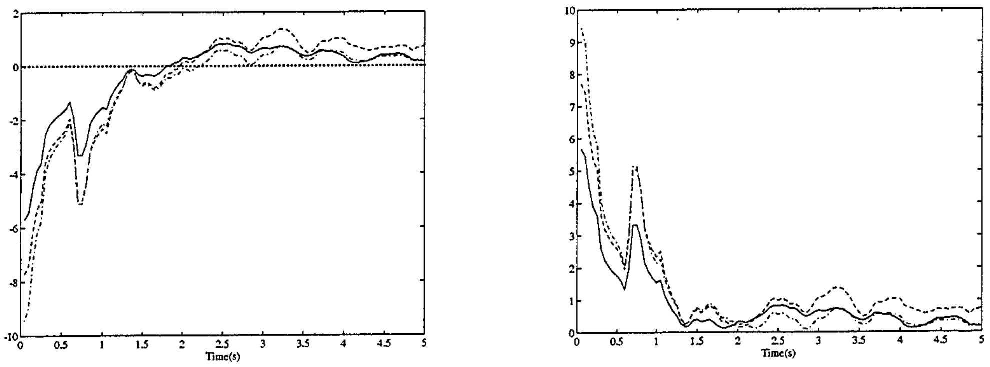


Figure 4.8: Vertical movements of distant points in the low signal to noise ratio case. Points 1 and 2 are displayed.



(a)

(b)

Figure 4.9: Bias and root mean-square error in the estimates of the 3D locations of close features, in the low signal to noise ratio case, from 50 Monte Carlo trials (solid line corresponds to the x coordinate, dashed line to the y coordinate, and dashed-dot line to the z coordinate). (a) Bias in Point 12, (b) root mean-square error in Point 12.

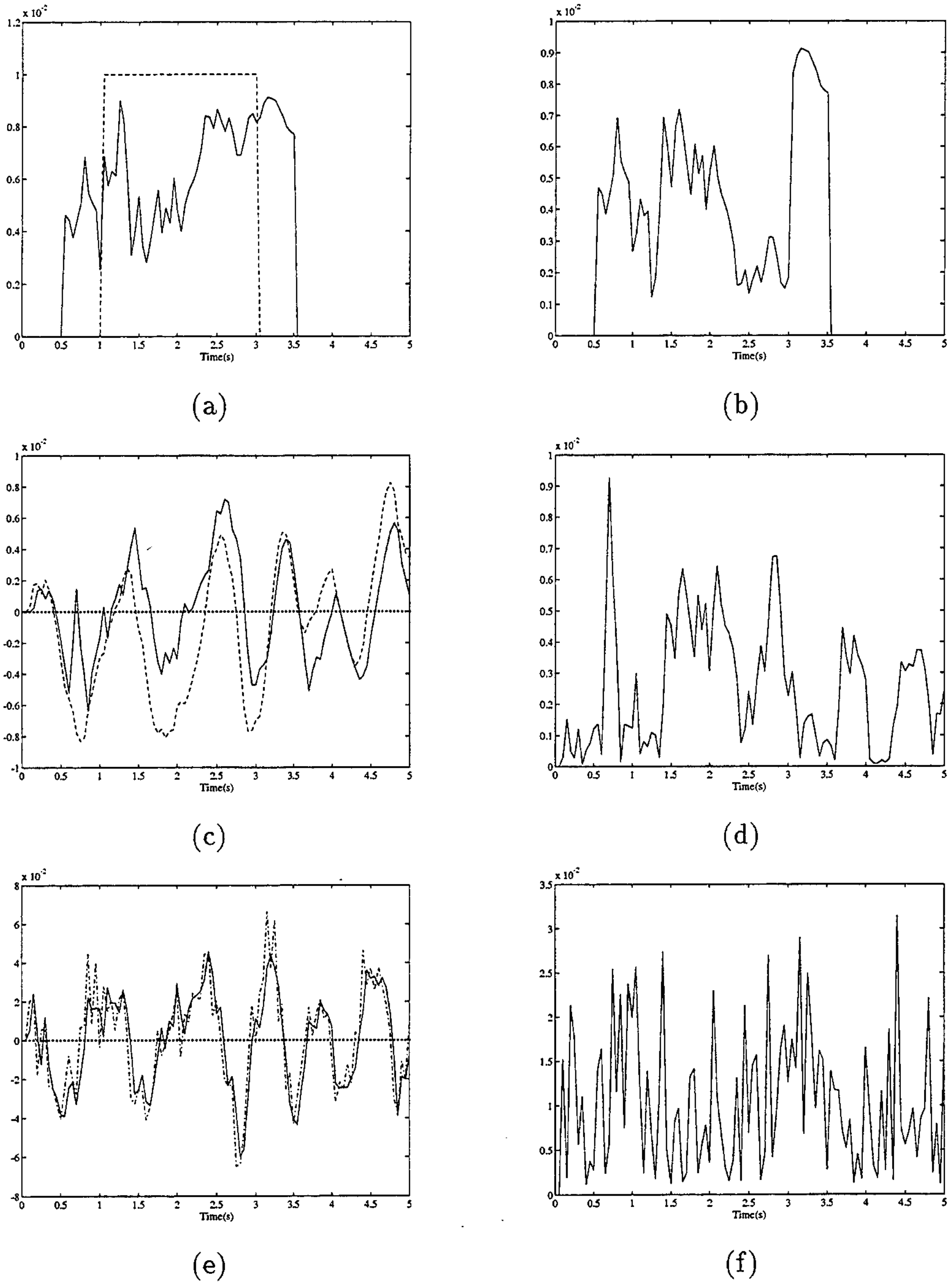


Figure 4.10: Estimates of climbing-pitch parameters, in the low signal to noise ratio case, from 50 Monte Carlo trials. (a) ω_{sy}: true (dashed line) and estimated (solid line) (b) ω_{sy}: root mean-square error, (c) θ: true (dashed line) and estimated (solid line) (d) θ: root mean-square error, (e) $\dot{\theta}$: true (dashed line) and estimated (solid line), (f) $\dot{\theta}$: root mean-square error.

Chapter 5

Feature Correspondence

5.1 Introduction

In this chapter, we describe a localized feature point tracking scheme. The merits of both two-view and multiple-view based methods, mentioned in Section 1.1.3, are considered in designing such an algorithm for finding trajectories of a set of feature points over a sequence. Basically, for each feature point of interest, a **local** 2D constant translational and rotational motion model is employed to exploit the temporal information in a sequence. To account for the non-smooth image motion due to 3D camera movements, a scheme is designed to estimate the inter-frame motion between two windows in which the point corresponding to the feature point of interest are likely to appear. Consequently, for each frame, the temporal information up to the particular frame is explicitly stored in a state vector; the state vector consists of the position of the corresponding point in the particular frame as well as the motion parameters between it and the previous frame. Since the state vector in each frame contains the accumulated information, tracking a feature point over a sequence is therefore decomposed into successive two-view matching problems. However, unlike two-view based approaches, the resulting problems are not independent.

Subsequently, to find the point corresponding to a feature point in the consecutive frame, a tracking algorithm consisting of three stages is performed. The first stage uses a Probabilistic Data Association Filter (PDAF) to obtain estimates of the 2D motion parameters between the two frames. The second stage employs techniques such as image warping, correlation matching, image differential estimation and bilinear interpolation to identify the corresponding point in the frame of interest. Finally, the third stage uses an EKF to update the state vector so that new temporal information can be processed.

In addition, when a sequence of images is considered, the scene in the images changes constantly. Feature points detected in the subsequent frames are therefore likely to provide more information for other higher-level applications such as obstacle avoidance, time-to-collision estimation, etc. A scheme is thus designed to include these newly extracted feature points. Therefore, the algorithm has the ability to track a dynamic set of feature points.

The organization of this chapter is as follows. The next section gives the trajectory model employed in this work. Section 5.3 presents the algorithm for finding corresponding points between two frames. A scheme for including new feature points is described in Section 5.4. Experimental results are reported in Section 5.5 and conclusions are given in Section 5.6.

5.2 Trajectory Model

There are, in general, two ways to describe the motion between two frames: 3D and 2D based methods. 3D based methods are based on an understanding of the camera motion while 2D based approaches apply 2D transformations. For the 3D based methods, since the motion of the camera needs to be estimated from the images, finding corresponding points is coupled with motion/structure estimation schemes. Because of the coupling of these two stages, errors in one stage are likely to affect the accuracy in the other stage. Besides, there are applications in which the motion information is not required while finding matching points is important [9]. It is therefore desirable to have a method independent of the 3D motion estimation scheme. Accordingly, 2D based approaches are considered.

Among 2D based techniques, three transformations: affine, projective and polynomial, are most commonly employed in a closely related problem, namely, image registration [8, 57]. Their appropriateness depends on the underlying motion of the camera. For convenience, let $\mathbf{p}_k : (x(k), y(k))^T$ denote a point in the k^{th} frame and $\mathbf{p}_{k+1} : (x(k+1), y(k+1))^T$ the resulting image point in the $(k+1)^{\text{st}}$ frame. Affine transformations account for rotation, translation, scaling and skew between two images as follows:

$$\mathbf{p}_{k+1} = \mathbf{A}_k \mathbf{p}_k + \mathbf{b}_k \tag{5.1}$$

where $\mathbf{A}_k$ and $\mathbf{b}_k$ are the transformation matrix and vector. Although affine transformations are easy to use, perspective distortions between images are not considered. To capture the image motion more closely, projective transformations can be used. They are expressed as

$$\mathbf{p}_{k+1} = (\mathbf{d}_k^T \mathbf{p}_k + 1)^{-1} (\mathbf{A}_k \mathbf{p}_k + \mathbf{b}_k) \tag{5.2}$$

where $\mathbf{d}_k$ specifies the transform coefficients for correcting the perspective distortion. Nonetheless, there still exist distortions not accounted for by affine and projective transformations [8, 57]. If even more satisfactory results are desired, nonlinear transformations such as polynomial transformations can be used. For example, a second order polynomial transformation can be written as

$$\mathbf{p}_{k+1} = \mathbf{C}_k \mathbf{p}_k \mathbf{p}_k^T \mathbf{d}_k + \mathbf{A}_k \mathbf{p}_k + \mathbf{b}_k \tag{5.3}$$

where the matrix $\mathbf{C}_k$ and vector $\mathbf{d}_k$ are the second order coefficients.

While the goal of image registration is to match two pictures **globally**, finding points corresponding to a feature point is, in essence, a **local** problem. The locality results from the perspective distortion as well as the camera motion. In this work, for a feature point of interest, a transformation is chosen to describe the image motion between two windows in which the matching points are likely to appear. Since only two windows are concerned, various distortions between the two windows are expected to be small. An affine transformation is therefore used. Moreover, to exploit the temporal information of a sequence, constant translational and rotational dynamics are assumed. Together, the image motion of the j^{th} feature point, between the k^{th} and $(k+1)^{\text{st}}$ frames, is modeled as

$$\mathbf{x}_j(k+1) = \mathbf{f}[\mathbf{x}_j(k)] + \mathbf{w}_j(k+1) \tag{5.4}$$

with

$$\mathbf{x}_j(k) = [x_j(k) \;\; y_j(k) \;\; t_{jx}(k) \;\; t_{jy}(k) \;\; \theta_j(k)]^T \tag{5.5}$$

and

$$\mathbf{f}[\mathbf{x}_j(k)] = \begin{pmatrix} x_j(k)\cos\theta_j(k) - y_j(k)\sin\theta_j(k) + t_{jx}(k) \\ x_j(k)\sin\theta_j(k) + y_j(k)\cos\theta_j(k) + t_{jy}(k) \\ t_{jx}(k) \\ t_jy(k) \\ \theta_j(k) \end{pmatrix} \tag{5.6}$$

where $(x_j(k), y_j(k))$ is the position of the corresponding point in the k^{th} frame. $t_{jx}(k)$, $t_{jy}(k)$, $\theta_j(k)$ respectively denote the associated translational movement along the x, y directions, and the rotation angle, all between the $(k-1)^{\text{st}}$ image and the k^{th} image; they are referred to as the motion parameters between the $(k-1)^{\text{st}}$ frame and the k^{th} frame. $\mathbf{w}_j(\cdot)$ is zero mean, white noise which is added to account for the deviation of the model from the stated assumption.

As discussed above, the consideration of a localized tracking scheme results in the choice of the simple affine transformation in this work. As shown in the experiments later, this model more or less captures the local image motion arising from arbitrary camera movements. It is therefore seen as a transformation between the 3D based tracking methods and the 2D global image registration techniques.

5.3 Feature Tracking between Two Frames

As mentioned earlier, tracking feature points over a sequence is decomposed temporally into successive two-view matching problems. Therefore, after the trajectory model has been chosen, we describe the tracking algorithm between two frames in this section. Without loss of generality, assuming that the j^{th} feature point has been tracked up to the k^{th} frame, the algorithm is described in terms of extending the trajectory to the $(k+1)^{\text{st}}$ frame. For clarity, an overview of the algorithm is first given and each step of the algorithm is then described successively.

5.3.1 Overview of the algorithm

To design an algorithm that combines the merits of both multiple-view and two-view based approaches, two issues are considered in devising a tracking scheme: how to exploit the temporal information such that the search area containing the corresponding point is small, and how to identify the corresponding point in the $(k+1)^{\text{st}}$ frame. In our work, an EKF/PDAF is employed to process the temporal information while a correlation-matching based technique identifies the corresponding point to sub-pixel accuracy. The resulting scheme therefore involves three stages: inter-frame motion estimation, corresponding point identification and temporal information filtering.

More specifically, the two-view tracking algorithm contains several steps summarized below:

- **Inter-frame motion estimation:** In the first stage, an inter-frame motion estimation scheme is applied. The scheme uses the accumulated temporal information up to the k^{th} frame to obtain an estimate of the motion between two neighborhoods which are likely to contain corresponding points of the j^{th} feature point.
- **Corresponding point identification**: The second stage identifies the corresponding point of the j^{th} feature point in the $(k+1)^{\text{st}}$ frame using the following procedures:
  1. Forward window warping and window extraction: Based on the inter-frame motion parameters, this step predicts the position of the corresponding point in the $(k+1)^{\text{st}}$

frame. An image warping technique is applied to generate a window centered at the predicted location from a neighborhood of the k^{th}-frame corresponding point. Also, another window possibly containing the corresponding point is extracted from the $(k+1)^{\text{st}}$ frame.

2. Grid neighbors matching: The second step employs a correlation-type matching method to find the corresponding point. However, the predicted location does not necessarily fall onto a grid location in the generated window. Alternatively, matching points of the four nearest grid neighbors are identified here.
3. Correct match verification: Before doing any further processing, if the corresponding point is less likely to be reliably identified based on the matches from the previous step, the algorithm stops tracking the feature point at this step.
4. Sub-pixel accuracy refinement: Once the algorithm thinks the corresponding point can be reliably found in the $(k+1)^{\text{st}}$ frame, this step applies an image-differential based scheme to improve the correlation matching results.
5. Matching point identification: Finally, in the last step, the algorithm uses a bilinear interpolation scheme to obtain the corresponding point in the $(k+1)^{\text{st}}$ frame.

- **Temporal information filtering**: The third stage processes the temporal information contained in the $(k+1)^{\text{st}}$ frame by updating the state vector $\mathbf{x}_j(k+1)$.

After these procedures are completed, the algorithm can continue tracking the j^{th} feature point to the $(k+2)^{\text{nd}}$ frame. We now describe each step of the two-view feature tracking algorithm in the following.

5.3.2 Inter-frame Motion Estimation

Temporal information contained in a sequence, in general, can be used to facilitate the feature tracking problem. To exploit the temporal information and accordingly, reduce the search area for finding the corresponding point, an inter-frame motion estimation scheme is devised. In this work, the PDAF [5] which was originally proposed for tracking a moving object in a cluttered environment is applied for this step. A cluttered environment, in the context of target tracking, refers to the situation where there exist ambiguities in the observations such that the designed tracking scheme does not know which observation is correct, or whether it is from the tracked object. Similarly, for tracking the j^{th} feature point, the ambiguities appear whenever there is more than one possible matching candidate in the $(k+1)^{\text{st}}$ frame.

For convenience, in addition to the trajectory model, define an observation model which relates the state vector $\mathbf{x}_j(k+1)$ with the noisy, but correct observation $\mathbf{z}_j(k+1)$ as follows

$$\mathbf{z}_j(k+1) = \mathbf{H}\mathbf{x}_j(k+1) + \mathbf{n}_j(k+1) \tag{5.7}$$

where $n_j(\cdot)$ is the zero mean, white observation noise,

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{5.8}$$

Based on the estimates of the state vector at the k^{th} frame $\hat{\mathbf{x}}_j(k|k)$,

$$\hat{\mathbf{x}}_j(k|k) = [\hat{x}_j(k|k)\ \hat{y}_j(k|k)\ \hat{t}_{jx}(k|k)\ \hat{t}_{jy}(k|k)\ \hat{\theta}_j(k|k)]^T \tag{5.9}$$

the inter-frame motion estimation scheme first predicts the $(k+1)^{\text{st}}$-frame state vector, $\tilde{\mathbf{x}}_j(k+1|k)$, according to (5.4):

$$\tilde{\mathbf{x}}_j(k+1|k) = \mathbf{f}[\hat{\mathbf{x}}_j(k|k)] \tag{5.10}$$

In particular, the predicted location of the corresponding point, $\tilde{\mathbf{z}}_j(k+1|k) = (\tilde{x}_j(k+1|k), \tilde{y}_j(k+1|k))$ is obtained as

$$\begin{aligned} \tilde{x}_j(k+1|k) &= \cos(\hat{\theta}_j(k|k))\hat{x}_j(k|k) - \sin(\hat{\theta}_j(k|k))\hat{y}_j(k|k) + \hat{t}_{jx}(k|k) \\ \tilde{y}_j(k+1|k) &= \sin(\hat{\theta}_j(k|k))\hat{x}_j(k|k) + \cos(\hat{\theta}_j(k|k))\hat{y}_j(k|k) + \hat{t}_{jy}(k|k) \end{aligned} \tag{5.11}$$

Subsequently, a window centered at $\tilde{\mathbf{z}}_j(k+1|k)$ is extracted from the $(k+1)^{\text{st}}$ frame and the feature point extraction algorithm reported in [36] is applied to the window to identify salient feature points. Due to the lack of information as well as the desire for higher accuracy, the corresponding point is not identified from the extracted points. Alternatively, a validation gate based on the Mahalanobis distance [13, 19] is constructed to select points for further processing. More specifically, define a validation gate centered at $\tilde{\mathbf{z}}_j(k+1|k)$ and with parameter γ as [5, 13]:

$$\mathbf{V}_{j,k+1}(\gamma) = \left\{\mathbf{z} : [\mathbf{z} - \tilde{\mathbf{z}}_j(k+1|k)]^T \mathbf{S}_j^{-1}(k+1)[\mathbf{z} - \tilde{\mathbf{z}}_j(k+1|k)] \leq \gamma\right\} \tag{5.12}$$

where $\mathbf{S}_j(k+1)$ is the covariance matrix of the innovation vector $\mathbf{z} - \tilde{\mathbf{z}}_j(k+1|k)$, and γ decides the scope of the validation gate. A set of extracted points is selected if their Mahalanobis distances are less than γ. Without loss of generality, it is assumed that there are $m_j(k+1)$ points, denoted as $\{\tilde{\mathbf{z}}_{j,i}(k+1), i = 1, \ldots, m_j(k+1)\}$, inside the validation gate $\mathbf{V}_{j,k+1}(\gamma)$. For clarity, Figure 5.1 shows a situation where five points have been extracted, the three points inside the validation gate will be selected.

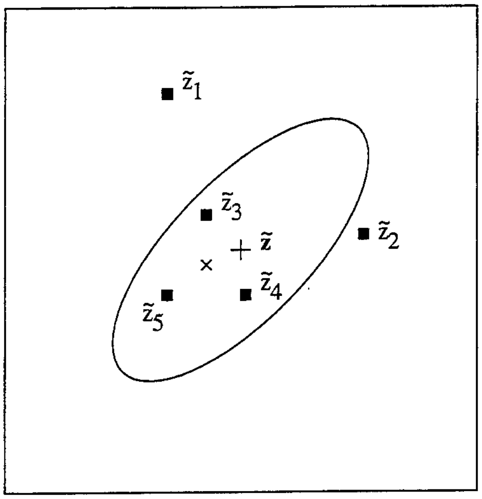


Figure 5.1: Illustrations of the PDAF: $\tilde{\mathbf{z}}$ is the predicted location and the ellipse represents the associated validation gate. $\{\tilde{\mathbf{z}}_i, i = 1, \ldots, 5\}$ are the five extracted feature points. Among them, $\{\tilde{\mathbf{z}}_3, \tilde{\mathbf{z}}_4, \tilde{\mathbf{z}}_5\}$ are inside the validation gate. The position marked by × corresponds to the position estimates computed by the PDAF.

Among the $m_j(k+1)$ extracted points, due to changes between images, each of the $m_j(k+1)$ points can be the corresponding point. In other words, if we consider these points as **noisy** corresponding points, we face a situation with ambiguities in observations. To process the information contained in these noisy observations, the PDAF is thereafter employed.

In essence, the PDAF obtains the information by associating different weights to different observations. A weight is assigned to an observation according to the a posteriori probability of an event which states that the corresponding observation is the correct one given the past data. As shown in [5], under the assumption that the observations are Normally distributed about $\tilde{\mathbf{z}}_j(k+1|k)$, these weights can be easily obtained. For convenience, denote these weights by $\{\beta_{j,i}(k+1), i = 0, 1, \ldots, m_j(k+1)\}$ respectively. Note that $\beta_{j,0}(k+1)$ is the weight assigned to the event that none of the observations are correct.

After each weight has been computed, the PDAF (or the inter-frame motion estimation scheme) processes the information from the $m_j(k+1)$ observations as follows

$$\tilde{\mathbf{x}}_j(k+1|k+1) = \tilde{\mathbf{x}}_j(k+1|k) + \tilde{\mathbf{K}}_j(k+1) \left[\sum_{i=1}^{m_j(k+1)} \beta_{j,i}(k+1)\tilde{\mathbf{v}}_{j,i}(k+1) \right] \tag{5.13}$$

where $\tilde{\mathbf{K}}_j(k+1)$ is the gain matrix defined similarly as the EKF,

$$\tilde{\mathbf{v}}_{j,i}(k+1) = \tilde{\mathbf{z}}_{j,i}(k+1) - \tilde{\mathbf{z}}_j(k+1|k) \tag{5.14}$$

For illustration purpose, the first two components of the state vector $\tilde{\mathbf{x}}_j(k+1|k+1)$, $(\tilde{x}_j(k+1|k+1), \tilde{y}_j(k+1|k+1))$, are supposedly marked by × in Figure 5.1. It is worth noting that if additional information is available such that the ambiguities in observations are resolved, then a PDAF is simplified to an EKF. This justifies the state vector prediction in (5.10).

Consequently, from (5.13), the estimates of motion parameters, i.e. the other three elements in $\tilde{\mathbf{x}}_j(k+1|k+1)$: $\tilde{t}_{jx}(k+1|k+1), \tilde{t}_{jy}(k+1|k+1), \tilde{\theta}_j(k+1|k+1)$, are obtained. We refer to them as the inter-frame motion parameters. Since the information in the $(k+1)^{\text{st}}$ frame is incorporated, the inter-frame motion parameters more or less capture the image motion of the neighborhoods of the j^{th} feature point due to the 3D movement of the camera. With these parameters available, the algorithm proceeds to the next stage to identify the corresponding point.

5.3.3 Corresponding Point Identification

In this stage, the corresponding point of the j^{th} feature point in the $(k+1)^{\text{st}}$ frame is found using a correlation matching approach followed by an interpolation scheme. The correlation matching method employed is similar to the block matching technique; however, inter-frame motion parameters are used so that the search area for the corresponding point is small and the matching results are more accurate. On the other hand, the interpolation scheme handles the problem due to the corresponding point not being at a grid location. This is important for tracking a feature point over a long sequence. Approximating the corresponding point by its nearest grid neighbor often leads to the situation that the corresponding point slowly moves away from the right position. The procedures used in this stage are presented in the following.

A. Forward Window Warping and Window Extraction The first step in identifying the corresponding point consists of obtaining two windows from the k^{th} and $(k+1)^{\text{st}}$ frames respectively. We refer to the window obtained from the k^{th} frame as the reference window $I_{j,r}$. The other one is called the target window $I_{j,t}$.

Consider $I_{j,r}$ first. Given the k^{th} frame and inter-frame motion parameters, we want to **predict** the reference window $I_{j,r}$. Similar problems have been studied in the area of image warping [57]. Basically, $I_{j,r}$ is an image that the algorithm believes the neighborhood of the corresponding point in the $(k+1)^{\text{st}}$ frame should look like. To obtain $I_{j,r}$, a more accurate prediction regarding the position of the corresponding point, $\hat{\mathbf{z}}_{j,r}(k+1|k) = (\hat{x}_{j,r}(k+1|k), \hat{y}_{j,r}(k+1|k))$, is used:

$$\begin{aligned} \hat{x}_{j,r}(k+1|k) &= \cos(\tilde{\theta}_j(k+1|k+1))\hat{x}_j(k) - \sin(\tilde{\theta}_j(k+1|k+1))\hat{y}_j(k) \\ &\quad +\tilde{t}_{jx}(k+1|k+1) \\ \hat{y}_{j,r}(k+1|k) &= \sin(\tilde{\theta}_j(k+1|k+1))\hat{x}_j(k) + \cos(\tilde{\theta}_j(k+1|k+1))\hat{y}_j(k) \\ &\quad +\tilde{t}_{jy}(k+1|k+1) \end{aligned} \tag{5.15}$$

where $\hat{\mathbf{z}}_j(k) = (\hat{x}_j(k), \hat{y}_j(k))$ is the position of the previously found k^{th} frame corresponding point. Note that $(\hat{x}_j(k), \hat{y}_j(k))$ is used instead of $(\hat{x}_j(k|k), \hat{y}_j(k|k))$. Subsequently, assume that pixels which are close to $(\hat{x}_j(k), \hat{y}_j(k))$ in the k^{th} frame undergo the same inter-frame motion, their locations in the $(k+1)^{\text{st}}$ frame are predicted in a similar way. After that, the predicted pixels are assigned with the same intensity values as the original pixels. This yields the reference window $I_{j,r}$ whose center is $\hat{\mathbf{z}}_{j,r}(k+1|k)$.

On the other hand, the target window $I_{j,t}$ is easier to obtain. By centering at $\hat{\mathbf{z}}_{j,r}(k+1|k)$, $I_{j,t}$ is directly extracted from the $(k+1)^{\text{st}}$ frame. To avoid potential confusion, we denote the center of $I_{j,t}$ as $\hat{\mathbf{z}}_{j,t}(k+1|k) = (\hat{x}_{j,t}(k+1|k), \hat{y}_{j,t}(k+1|k))$, although $\hat{\mathbf{z}}_{j,t}(k+1|k)$ has the same value as $\hat{\mathbf{z}}_{j,r}(k+1|k)$.

Since $I_{j,r}$ is obtained with the knowledge of the inter-frame motion, a correlation matching method can be employed to find the corresponding point. Thus, given $I_{j,r}$ and $I_{j,t}$, if $\hat{\mathbf{z}}_{j,r}(k+1|k)$ happens to be a grid point, then the corresponding point of the j^{th} feature point can be directly found. However $\hat{\mathbf{z}}_{j,r}(k+1|k)$, in general, is not located at a grid point. We describe how to handle this situation next.

B. Grid Neighbors Matching Correlation matching techniques typically match a grid point in one image into a grid point in another image. In our case, the problem appears when the predicted location $\hat{\mathbf{z}}_{j,r}(k+1|k)$ does not coincide with a grid point. A scheme suggested in [61] is employed to solve this problem. The approach is basically composed of two steps: (1). Find matching points, in $I_{j,t}$, of the four nearest grid points of $\hat{\mathbf{z}}_{j,r}(k+1|k)$ using correlation matching methods. (2). Interpolate the results to obtain the desired corresponding point. We describe the first step here. The second step is discussed later.

For convenience, denote the four nearest grid points of $\hat{\mathbf{z}}_{j,r}(k+1|k)$ by $\{\hat{\mathbf{z}}_{j,i}(k+1|k), i = 1, \ldots, 4\}$,

$$\begin{aligned} \hat{\mathbf{z}}_{j,1}(k+1|k) &= ([\hat{x}_{j,r}(k+1|k)],\ [\hat{y}_{j,r}(k+1|k)]) \\ \hat{\mathbf{z}}_{j,2}(k+1|k) &= ([\hat{x}_{j,r}(k+1|k)],\ [\hat{y}_{j,r}(k+1|k)]+1) \\ \hat{\mathbf{z}}_{j,3}(k+1|k) &= ([\hat{x}_{j,r}(k+1|k)]+1,\ [\hat{y}_{j,r}(k+1|k)]) \\ \hat{\mathbf{z}}_{j,4}(k+1|k) &= ([\hat{x}_{j,r}(k+1|k)]+1,\ [\hat{y}_{j,r}(k+1|k)]+1) \end{aligned} \tag{5.16}$$

where $[\cdot]$ represents the floor function which rounds a real number to the nearest smaller integer. Without loss of generality, we focus on finding the matching point of $\hat{\mathbf{z}}_{j,1}(k+1|k)$ here. The matching points of the other three grid points can be found similarly.

To employ the correlation matching method, a template $\Omega_{I_{j,r}}(\hat{\mathbf{z}}_{j,1}(k+1|k))$ centered at $\hat{\mathbf{z}}_{j,1}(k+1|k)$ is first created from $I_{j,r}$. Then, in $\Omega_{I_{j,r}}(\hat{\mathbf{z}}_{j,1}(k+1|k))$, to put more emphasis on pixels close to $\hat{\mathbf{z}}_{j,1}(k+1|k)$, each pixel is further assigned with a different weight according to the following:

$$\mu_{l,m} = \begin{cases} 1 & \text{if}(l,m) = (0,0) \\ \frac{c}{8\max(|l|,|m|)} & \text{otherwise} \end{cases} \tag{5.17}$$

where c is a constant and (l,m) are integers such that $([\hat{x}_{j,r}(k+1|k)]+l,[\hat{y}_{j,r}(k+1|k)]+m) \in \Omega_{I_{j,r}}(\hat{\mathbf{z}}_{j,1}(k+1|k))$. The resulting template is thereupon seen as a weighted template. Note that for any $\mathbf{q} \in I_{j,t}$, a template $\Omega_{I_{j,t}}(\mathbf{q})$ centered at $\mathbf{q}$ with similar weight assignments can also be obtained.

Subsequently, if we employ the weighted templates and define the similarity measure between $\mathbf{p} = (p_1,p_2) \in I_{j,r}$ and $\mathbf{q} = (q_1,q_2) \in I_{j,t}$ as

$$\psi_{I_{j,r}I_{j,t}}(\mathbf{p},\mathbf{q}) \equiv \frac{C(\mathbf{p},\mathbf{q})}{N_1(\mathbf{p}) \cdot N_2(\mathbf{q})} \tag{5.18}$$

where $g_1(\cdot), g_2(\cdot)$ are the intensity functions of the pixels in $I_{j,r}$ and $I_{j,t}$ respectively, and

$$\begin{aligned}
C(\mathbf{p},\mathbf{q}) &= \sum_{l,m} \mu_{lm}[g_1(p_1+l,p_2+m)-\bar{g}_1(\mathbf{p})]\cdot[g_2(q_1+l,q_2+m)-\bar{g}_2(\mathbf{q})] && (5.19)\\
N_1(\mathbf{p}) &= \sqrt{\sum_{l,m} \mu_{lm}[g_1(p_1+l,p_2+m)-\bar{g}_1(\mathbf{p})]^2} && (5.20)\\
N_2(\mathbf{q}) &= \sqrt{\sum_{l,m} \mu_{lm}[g_2(q_1+l,q_2+m)-\bar{g}_2(\mathbf{q})]^2} && (5.21)\\
\bar{g}_1(\mathbf{p}) &= \frac{1}{N}\sum_{l,m} g_1(p_1+l,p_2+m) && (5.22)\\
\bar{g}_2(\mathbf{q}) &= \frac{1}{N}\sum_{l,m} g_2(q_1+l,q_2+m) && (5.23)
\end{aligned}$$

with N being the number of pixels in both $\Omega_{I_{j,r}}(\mathbf{p})$ and $\Omega_{I_{j,t}}(\mathbf{q})$, then the matching point of $\hat{\mathbf{z}}_{j,1}(k+1|k)$ is obtained by searching for a grid point in $I_{j,t}$ which has the highest similarity measure with $\hat{\mathbf{z}}_{j,1}(k+1|k)$. (For example, $-5 \leq l,m \leq 5$ and $c = 2$ are used in our experiments.)

A few remarks regarding the above correlation matching scheme are given in the following. First, the similarity measure (5.18) has the same property as the well-known correlation coefficient [61]:

$$|\psi_{I_{jr}I_{jt}}| \leq 1 \tag{5.24}$$

Second, the use of a weighted template is expected to achieve better localization because of the imperfect feature detection scheme. By this, we mean that the extracted feature points usually are not located exactly at the corners due to quantization effects. Typically, they are a few pixels away from the boundaries. Experiments show that giving higher weights to pixels near the center of the template helps in achieving better localization later. Third, since the inter-frame motion has been considered, to find the matching grid point of $\hat{\mathbf{z}}_{j,1}(k+1|k)$, only a small area centered at $([\hat{x}_{j,t}(k+1|k)],[\hat{y}_{j,t}(k+1|k)]+1)$ in $I_{j,t}$ needs to be searched. (In particular, a 10×10 window is searched in the experiments.) Figure 5.2 illustrates a possible configuration obtained in this step. For convenience, we denote the resulting matching grid points of $\{\hat{\mathbf{z}}_{j,i}(k+1), i = 1,\ldots,4\}$ by $\{\mathbf{y}_{j,i}(k+1), i = 1,\ldots,4\}$.

Up to now, we have implicitly assumed that the point corresponding to the j^{th} feature point exists in the $(k+1)^{\text{st}}$ frame, and the neighborhoods of corresponding points in the k^{th} and $(k+1)^{\text{st}}$ frames are similar. However, there exist situations where these assumptions may not be valid. We next discuss a scheme which verifies these assumptions.

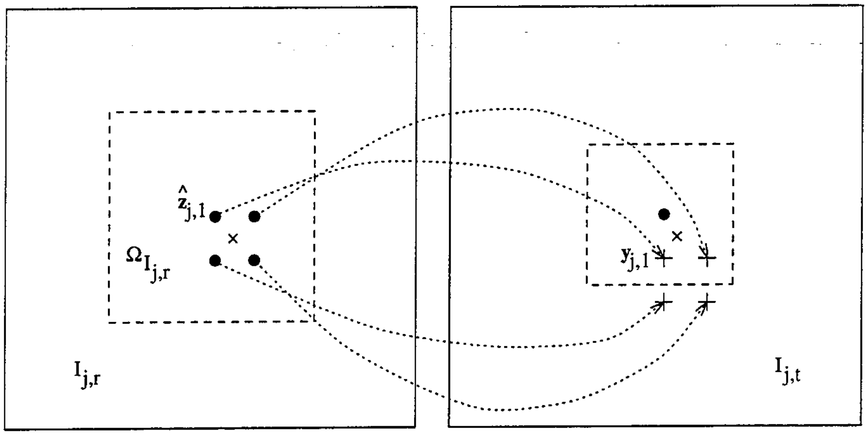


Figure 5.2: Illustrations of grid neighbors matching: $\hat{\mathbf{z}}_{j,r}, \hat{\mathbf{z}}_{j,t}$ are marked by × in $I_{j,r}$ and $I_{j,t}$ respectively. The dashed square in $I_{j,r}$ represents the template $\Omega_{I_{j,r}}(\hat{\mathbf{z}}_{j,1})$. The dashed square in $I_{j,t}$ is the search area where the matching point of $\hat{\mathbf{z}}_{j,1}$, $\mathbf{y}_{j,1}$, is to be searched for. The matching points of all four grid neighbors of $\hat{\mathbf{z}}_{j,r}$ are marked by + in $I_{j,t}$.

C. Correct Match Verification It is well known that when a feature point is tracked to the $(k+1)^{\text{st}}$ frame, the corresponding point may be occluded. Even if the corresponding point is not occluded, the relative structures in the neighborhoods of corresponding points in the k^{th} and $(k+1)^{\text{st}}$ frames may change dramatically. Both situations lead to incorrect matches using the previously described correlation matching method. Therefore, before advancing to the next step, the hypothesis that the matches $\{(\hat{\mathbf{z}}_{j,i}(k+1|k), \mathbf{y}_{j,i}(k+1)), i = 1, \ldots, 4\}$ are correct is tested here.

We select a threshold, say TH, as follows, to decide whether $(\hat{\mathbf{z}}_{j,i}(k+1|k), \mathbf{y}_{j,i}(k+1))$ is a correct match for each $i \in \{1, \ldots, 4\}$:

$$\begin{cases} \psi_{I_{j,r}I_{j,t}}(\hat{\mathbf{z}}_{j,i}(k+1|k), \mathbf{y}_{j,i}(k+1)) \geq TH & (\hat{\mathbf{z}}_{j,i}(k+1|k), \mathbf{y}_{j,i}(k+1)) \text{ is correct} \\ \psi_{I_{j,r}I_{j,t}}(\hat{\mathbf{z}}_{j,i}(k+1|k), \mathbf{y}_{j,i}(k+1)) < TH & \text{otherwise} \end{cases} \tag{5.25}$$

For convenience, denote the number of correct matches among $\{(\hat{\mathbf{z}}_{j,i}(k+1|k), \mathbf{y}_{j,i}(k+1)), i = 1, \ldots, 4\}$ by n. Three cases are considered in the following:

Case 1: $n \leq 2$

In this case, more than two matches of grid neighbors of $\hat{\mathbf{z}}_{j,r}(k+1|k)$ are likely to be incorrect. Consequently, the algorithm feels that the real point corresponding to the j^{th} feature point cannot be reliably found in the $(k+1)^{\text{st}}$ frame. No further tracking will be attempted.

Case 2: $n = 3$

In this case, one of the matches is regarded as unreliable. Without loss of generality, we assume that $\mathbf{y}_{j,1}(k+1)$ is the unreliable matching point. Since there are three other correct matches, the algorithm tries to correct the wrong match using an extrapolation scheme. More specifically, because $\{\hat{\mathbf{z}}_{j,i}(k+1|k), i = 1, \ldots, 4\}$ form a square in $I_{j,r}$, the extrapolation scheme assumes that $\{\mathbf{y}_{j,i}(k+1), i = 1, \ldots, 4\}$ should form a parallelogram in $I_{j,t}$. The matching point of $\hat{\mathbf{z}}_{j,1}(k+1|k)$ is then replaced by

$$\mathbf{y}_{j,1}(k+1) = \mathbf{y}_{j,2}(k+1) + \mathbf{y}_{j,3}(k+1) - \mathbf{y}_{j,4}(k+1) \tag{5.26}$$

For clarity, we illustrate this procedure in Figure 5.3. Assume that $\mathbf{y}_{j,1}(k+1)$ is originally matched to the conceivably wrong point marked by + due to changes in the neighborhood. Applying (5.26) yields another matching point for $\hat{\mathbf{z}}_{j,1}(k+1|k)$ as indicated in the figure.

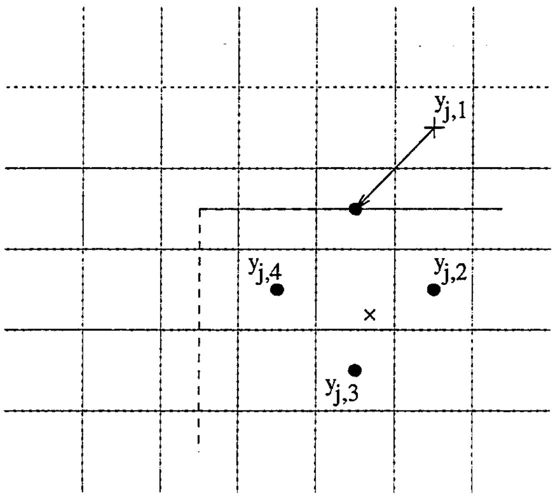


Figure 5.3: Illustrations of case 2 of correct match verification: × denotes the real corresponding point. The original $\mathbf{y}_{j,1}$ is denoted by + while the extrapolated one is denoted by •.

Case 3: $n = 4$

In this case, $\{(\hat{\mathbf{z}}_{j,i}(k+1|k), \mathbf{y}_{j,i}(k+1)), i = 1, \ldots, 4\}$ are all confirmed to be correct.

Among three cases, if either case 2 or case 3 applies, the algorithm proceeds to the next step.

D. Sub-pixel Accuracy Refinement Since $\{(\hat{\mathbf{z}}_{j,i}(k+1|k), i = 1, \ldots, 4\}$ are only matched to grid-level accuracy by the correlation matching scheme, this step tries to improve the matching accuracy to the sub-pixel level. In our work, an image-differential based technique [27] is employed for the purpose [61]. Again, we illustrate this scheme in terms of $(\hat{\mathbf{z}}_{j,1}(k+1|k), \mathbf{y}_{j,1}(k+1))$.

Recall that $g_2(\cdot)$ is the intensity function of pixels in $I_{j,t}$. Under the assumption that $g_2(\cdot)$ has an offset $\mathbf{\Delta} = (\delta_x, \delta_y)$ relative to another intensity function $g_2'(\cdot)$ at $\mathbf{y}_{j,1}(k+1)$:

$$g_2(\mathbf{y}_{j,1}(k+1)) = g_2'(\mathbf{y}_{j,1}(k+1) + \mathbf{\Delta}) \tag{5.27}$$

the image differential scheme approximates the difference between $g_2(\mathbf{y}_{j,1}(k+1))$ and $g_2'(\mathbf{y}_{j,1}(k+1))$, $d(\mathbf{y}_{j,1}(k+1))$, using the first-order Taylor series expansion:

$$\begin{aligned} d(\mathbf{y}_{j,1}(k+1)) &\equiv g_2(\mathbf{y}_{j,1}(k+1)) - g_2'(\mathbf{y}_{j,1}(k+1)) & (5.28)\\ &\approx \ < \nabla g_2(\mathbf{y}_{j,1}(k+1)), \mathbf{\Delta} > & (5.29) \end{aligned}$$

where $< \cdot, \cdot >$ represents the inner product of the arguments, and $\nabla g_2(\mathbf{y}_{j,1}(k+1))$ is the gradient vector,

$$\nabla g_2(\mathbf{y}_{j,1}(k+1)) \equiv \left(\frac{\partial g_2(\mathbf{y}_{j,1}(k+1))}{\partial x}, \frac{\partial g_2(\mathbf{y}_{j,1}(k+1))}{\partial y} \right)^T . \tag{5.30}$$

Subsequently, assume that the approximation (5.29) holds for a small neighborhood around $\mathbf{y}_{j,1}(k+1)$ of size $(2\omega_d + 1) \times (2\omega_d + 1)$; then a set of equations can be found:

$$\mathbf{G\Delta} = \mathbf{D} \tag{5.31}$$

where

$$\begin{aligned} \mathbf{G} &= [\nabla g_2(\mathbf{y}_{j,1}(k+1)+u_{-\omega_d,-\omega_d}),\ldots,\nabla g_2(\mathbf{y}_{j,1}(k+1)+u_{0,0}), \\ &\quad \ldots,\nabla g_2(\mathbf{y}_{j,1}(k+1)+u_{\omega_d,\omega_d})]^T \qquad (5.32) \\ \mathbf{D} &= [d(\mathbf{y}_{j,1}(k+1)+u_{-\omega_d,-\omega_d}),\ldots,d(\mathbf{y}_{j,1}(k+1)+u_{0,0}), \\ &\quad \ldots,d(\mathbf{y}_{j,1}(k+1)+u_{\omega_d,\omega_d})]^T \qquad (5.33) \end{aligned}$$

with

$$u_{\nu_1,\nu_2}=(\nu_1,\nu_2)^T,\ \ -\omega_d\le\nu_1,\nu_2\le\omega_d \qquad (5.34)$$

such that $(\mathbf{y}_{j,1}(k+1)+u_{\nu_1,\nu_2})$ is in the $(2\omega_d+1)\times(2\omega_d+1)$ neighborhood of $\mathbf{y}_{j,1}(k+1)$.

Afterwards, the offset vector $\mathbf{\Delta}$ is obtained from (5.31) using a least square approach. (In our experiments, $\omega_d=3$.) $\hat{\mathbf{z}}_{j,1}(k+1|k)$ is then matched to $\mathbf{y}'_{j,1}(k+1)\equiv(x'_{j,1}(k+1),y'_{j,1}(k+1))=(\mathbf{y}_{j,1}(k+1)+\mathbf{\Delta})$. Similarly, the sub-pixel matching of the other three grid points can also be obtained. We denote these four sub-pixel matching points by $\{\mathbf{y}'_{j,i}(k+1),i=1,\ldots,4\}$.

E. Matching Point Identification After $\{\mathbf{y}'_{j,i}(k+1),i=1,\ldots,4\}$ have been obtained, the algorithm identifies the corresponding point of the j^{th} feature point using a bilinear interpolation scheme in this step.

The scheme uses matches $\{(\hat{\mathbf{z}}_{j,i}(k+1|k),\mathbf{y}'_{j,i}(k+1)),i=1,\ldots,4\}$ to find the corresponding point of the j^{th} feature point. More specifically, assume that for any pixel $\mathbf{p}=(p_1,p_2)\in I_{j,r}$ which is inside or on the boundary of the square formed by $\{\hat{\mathbf{z}}_{j,i}(k+1|k),i=1,\ldots,4\}$, its matching point in $I_{j,t}$, $\mathbf{q}=(q_1,q_2)$, can be obtained as

$$\left\{\begin{array}{lcl} q_1 &=& \alpha_1+\alpha_2p_1+\alpha_3p_2+\alpha_4p_1p_2 \\ q_2 &=& \beta_1+\beta_2p_1+\beta_3p_2+\beta_4p_1p_2 \end{array}\right. \qquad (5.35)$$

where $\{\alpha_i,\beta_i,i=1,\ldots,4\}$ are constant coefficients to be found.

Then, since the matching points of $\{\hat{\mathbf{z}}_{j,i}(k+1|k),i=1,\ldots,4\}$ have been obtained, $\{\alpha_i,\beta_i,i=1,\ldots,4\}$ can be found easily [57, 61]. Accordingly, using these coefficients and (5.35), the corresponding point of the j^{th} feature point $\hat{\mathbf{z}}_j(k+1)=(\hat{x}_j(k+1),\hat{y}_j(k+1))$, are computed by

$$\left\{\begin{array}{lcl} \hat{\mathbf{z}}_j(k+1) &=& \mathbf{y}'_{j,1}(k+1)+\epsilon_x[\mathbf{y}'_{j,3}(k+1)-\mathbf{y}'_{j,1}(k+1)]+\epsilon_y[\mathbf{y}'_{j,2}(k+1)-\mathbf{y}'_{j,1}(k+1)] \\ && +\epsilon_x\epsilon_y[\mathbf{y}'_{j,1}(k+1)+\mathbf{y}'_{j,4}(k+1)-\mathbf{y}'_{j,2}(k+1)-\mathbf{y}'_{j,3}(k+1)] \\ \epsilon_x &=& \hat{x}_{j,r}(k+1|k)-[\hat{x}_{j,r}(k+1|k)] \\ \epsilon_y &=& \hat{y}_{j,r}(k+1|k)-[\hat{y}_{j,r}(k+1|k)] \end{array}\right. \qquad (5.36)$$

where $(\hat{x}_{j,r}(k+1|k),\hat{y}_{j,r}(k+1|k))$ and $([\hat{x}_{j,r}(k+1|k)],[\hat{y}_{j,r}(k+1|k)])$ have been defined in (5.15) and (5.16) respectively.

This ends the stage of corresponding point identification. The algorithm then advances to the next stage of temporal information filtering.

5.3.4 Temporal Information Filtering

To facilitate the tracking scheme, the temporal information between the k^{th} and $(k+1)^{\text{st}}$ frames is processed. As mentioned earlier, accumulated temporal information up to the $(k+1)^{\text{st}}$ frame is stored in $\mathbf{x}_j(k+1)$ in our work. For convenience, we repeat the trajectory model (5.4) and the observation model (5.7) here:

$$\begin{aligned}\mathbf{x}_j(k+1) &= \mathbf{f}[\mathbf{x}_j(k)] + \mathbf{w}_j(k+1)\\ \mathbf{z}_j(k+1) &= \mathbf{H}\mathbf{x}_j(k+1) + \mathbf{n}_j(k+1).\end{aligned}$$

Then, given the corresponding point $\hat{\mathbf{z}}_j(k+1)$, the current stage employs an EKF to obtain the estimate of $\mathbf{x}_j(k+1)$, $\hat{\mathbf{x}}_j(k+1|k+1)$.

Since $\mathbf{f}$ is nonlinear, in order to apply the EKF, the following matrix is first defined:

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}. \tag{5.37}$$

The temporal information is then processed by performing following steps successively:

- **Step 1:** State and covariance propagation

$$\begin{aligned}\hat{\mathbf{x}}_j(k+1|k) &= \mathbf{f}[\hat{\mathbf{x}}_j(k|k)]\\ \hat{\mathbf{P}}_j(k+1|k) &= \mathbf{F}[\hat{\mathbf{x}}_j(k|k)]\hat{\mathbf{P}}_j(k|k)\mathbf{F}[\hat{\mathbf{x}}_j(k|k)]^T + \mathbf{Q}_j(k+1)\end{aligned} \tag{5.38}$$

  where $\hat{\mathbf{x}}_j(k|k)$ is defined in (5.9). $\hat{\mathbf{P}}_j(k|k), \hat{\mathbf{P}}_j(k+1|k), \mathbf{Q}_j(k+1)$ are covariance matrices of $\hat{\mathbf{x}}_j(k|k), \hat{\mathbf{x}}_j(k+1|k)$ and $\mathbf{w}_j(k+1)$ respectively.

- **Step 2:** State and covariance update

$$\begin{aligned}\mathbf{K}_j(k+1) &= \hat{\mathbf{P}}_j(k+1|k)\mathbf{H}^T[\mathbf{H}\hat{\mathbf{P}}_j(k+1|k)\mathbf{H}^T + \mathbf{R}_j(k+1)]^{-1}\\ \hat{\mathbf{x}}_j(k+1|k+1) &= \hat{\mathbf{x}}_j(k+1|k) + \mathbf{K}_j(k+1)[\hat{\mathbf{z}}_j(k+1) - \mathbf{H}\hat{\mathbf{x}}_j(k+1|k)]\\ \hat{\mathbf{P}}_j(k+1|k+1) &= [I - \mathbf{K}_j(k+1)\mathbf{H}]\hat{\mathbf{P}}_j(k+1|k)\end{aligned} \tag{5.39}$$

  where $\mathbf{R}_j(k+1)$ is the covariance matrix of $\mathbf{n}_j(k+1)$, and I is the identity matrix. $\mathbf{K}_j(k+1)$ represents the gain matrix, while $\hat{\mathbf{x}}_j(k+1|k+1)$ is the updated state vector and $\hat{\mathbf{P}}_j(k+1|k+1)$ is the associated covariance matrix.

Note that in applying the EKF to process the temporal information, the corresponding point $\hat{\mathbf{z}}_j(k+1)$ is assumed to be normally distributed around $\mathbf{H}\hat{\mathbf{x}}_j(k+1|k)$. This is a convenient assumption, and it also works well in our experiments. Moreover, $\hat{\mathbf{x}}_j(k+1|k+1)$, in particular the position estimates $(\hat{x}_j(k+1|k+1), \hat{y}_j(k+1|k+1))$, are only used to exploit the temporal information as shown in (5.11). The observation $\hat{\mathbf{z}}_j(k+1)$ should be used as the corresponding point of the j^{th} feature point in the $(k+1)^{\text{st}}$ frame whenever higher-level applications are considered.

After the temporal information is processed, the task of tracking the j^{th} feature point to the $(k+1)^{\text{st}}$ frame is completed. The algorithm can now continue tracking the j^{th} feature point to the $(k+2)^{\text{nd}}$ frame. Before presenting the experimental results of the tracking algorithm, we digress briefly to present a scheme for including new feature points detected in the $(k+1)^{\text{st}}$ frame in the next section.

5.4 Inclusion of New Features

When tracking a set of feature points over a sequence, it is likely that some points disappear after some frames. Accordingly, related tasks may be affected since the amount of information is reduced. (Here, each tracked feature point is assumed to provide useful information, although in some applications, only a minimum number of point correspondences is required.) In addition, since the scene in the sequence constantly changes, it is desirable to track feature points in the successive frames which may well contain more information than feature points currently being tracked. We suggest a scheme which considers the problem of including new feature points detected later in the sequence.

Assume that the algorithm has already completed tracking all feature points from the k^{th} frame to the $(k+1)^{\text{st}}$ frame. Without loss of generality, suppose that there remain M feature points in the $(k+1)^{\text{st}}$ frame and consequently, there exist M validation gates, $\{\mathbf{V}_{1,k+1}(\gamma), \mathbf{V}_{2,k+1}(\gamma), \ldots, \mathbf{V}_{M,k+1}(\gamma)\}$. Note that the M validation gates are defined analogously to (5.12). Then since $\mathbf{V}_{j,k+1}(\gamma)$ represents a neighborhood of the corresponding point of the j^{th} feature point, we assume that points inside $\mathbf{V}_{j,k+1}(\gamma)$ more or less carry the same information as the j^{th} feature point for other applications. Equivalently, for points extracted from the $(k+1)^{\text{st}}$ frame by the algorithm reported in [36], only those points outside all of the M validation gates are regarded as new feature points. The tracking algorithm therefore starts tracking these points only.

For example, consider Figure 5.4. Assume that the tracking algorithm maintains eight trajectories in the $(k+1)^{\text{st}}$ frame. This results in eight validation gates. Meanwhile, the feature extraction algorithm extracts nine feature points from the $(k+1)^{\text{st}}$ frame. Since only $\mathbf{z}'_1, \mathbf{z}'_6$ and $\mathbf{z}'_9$ are outside all the eight validation gates, the algorithm only recognizes these three as new feature points.

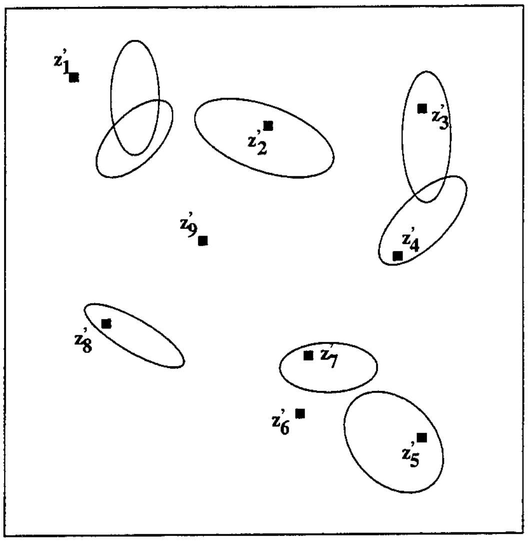


Figure 5.4: Illustrations of the scheme of including new feature points: The tracking algorithm tracks eight feature points to the $(k+1)^{\text{st}}$ frame and forms eight validation gates. $\mathbf{z}'_1, \ldots, \mathbf{z}'_9$ are nine newly extracted feature points. Only $\mathbf{z}'_1, \mathbf{z}'_6$ and $\mathbf{z}'_9$ are considered as new feature points in the $(k+1)^{\text{st}}$ frame.

As shown in the experiments later, the proposed scheme is quite efficient. It not only handles the problem of the decreasing number of tracked feature points, it also prevents the number of feature points from growing too fast. This is because that when more feature points are tracked, the image region covered by the associated validation gates also enlarges.

This completes the description of the scheme for including new feature points as well as our algorithm for tracking a dynamic set of feature points. The next section presents experimental results on four real image sequences.

5.5 Experimental Results

In this section, tracking results are presented for four real image sequences taken by cameras undergoing different types of motion. A tracking list which contains the corresponding points as well as the new feature points is created and updated at every frame. For visual purposes, only the trajectories of the feature points tracked from the first frame as well as the new feature points added to the tracking list at subsequent frames are displayed. The dynamic behavior of the algorithm is shown in a table which lists the number of feature point trajectories being maintained or removed from the tracking list and the number of new points selected from every frame.

5.5.1 UMASS PUMA2 Sequence

The first sequence is known as the UMASS PUMA2 sequence; it consists of thirty 256 × 256 frames. The camera is connected to the end of a PUMA robot arm and rotates about a rotation center which is close to the image center. Figure 5.5 shows the trajectories for a set of feature points automatically extracted from the first frame by the algorithm reported in [36]; the trajectories are shown up to the 19th and 30th frames. The new feature points extracted by the feature extraction algorithm from frames 3, 19 and 30 are also shown in Figure 5.5 in addition to the labeled points which were added to the tracking list at different time instants. The number of feature points being tracked varies with time, as shown in Table 5.1. As seen in Table 5.1, the algorithm for adding new points to the tracking list efficiently maintains the number of points on the list.

Table 5.1: The number of feature points in the tracking list for the UMASS PUMA2 Sequence

frame number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
tracked points	0	21	19	28	31	35	38	40	41	40	41	43	43	44	46	49	48	49	48	50	49	51	50	53	50	52	52	53	54	53
extracted points	23	23	22	26	24	27	29	29	28	28	25	16	19	20	24	18	21	20	21	20	21	20	19	23	25	24	24	27	24	15
new points	23	0	9	4	5	5	3	3	4	4	3	1	2	2	5	2	3	1	3	1	3	0	4	0	3	0	3	2	1	1

5.5.2 Coke Can Sequence

The second sequence is the Coke Can Sequence, in which the camera is approaching the scene, with the Focus of Expansion(FOE) located on the coke can. Fifteen frames chosen from the densely sampled sequence (every tenth frame) are used. The original 512 × 512 images are down-sampled to 256 × 256 before applying the algorithm. The resulting trajectories from the first frame to the 10th and 15th frames are shown in Figure 5.6. As seen from the figures, because of the pure translation of the camera, the trajectories of the feature points diverge from the FOE. The number of tracked feature points at each time instant is listed in Table 5.2. The new feature points added at the 2nd, 10th, and 15th frames are also marked in Figure 5.6.

Table 5.2: The number of feature points in the tracking list for the Coke Can Sequence

frame number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
tracked points	0	12	13	16	18	20	22	22	22	26	27	28	28	28	28
extracted points	13	12	15	12	13	17	15	15	16	15	14	17	15	17	14
new points	13	1	4	2	2	2	1	1	5	1	1	2	0	1	1

5.5.3 Rocket ALV Sequence

The third sequence is the 30-frame UMASS Rocket ALV Sequence. Again, the 512 × 512 images are down-sampled to 256 × 256 before applying the algorithm. In this sequence, the camera is mounted on a vehicle which appears to be moving along a straight line to the left and into the image plane with almost no rotation. Due to the uneven terrain, the motion of the camera is not smooth. The trajectories for the feature points up to the 13th and 30th frames are shown in Figure 5.7. Table 5.3 lists the number of feature points on the tracking list. It is noted that, for outdoor images acquired from a moving vehicle, the scene close to the camera normally appears in the lower part of the images. A threshold is therefore set to remove the detected points which are far away such as points on the cloud. The extracted feature points as well as the new points selected by the criterion in Section 5.4 from the 2nd, 13th and 30th frames are also shown in Figure 5.7. In this sequence, many feature points move out of the field of view in the first few frames. It is therefore necessary to include new feature points when they become available. Also, it is apparent from the sequence that the vehicle has an abrupt change in heading direction at the 16th and 20th frames, but the algorithm still keeps tracking most of the feature points.

Table 5.3: The number of feature points in the tracking list for the UMASS Rocket ALV Sequence

frame number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
tracked points	0	19	19	23	22	22	19	16	19	21	24	22	22	23	21
extracted points	25	20	16	12	14	13	16	16	13	18	14	21	16	17	14
new points	25	4	6	1	3	0	7	8	6	6	2	2	3	1	1
frame number	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
tracked points	19	19	18	18	21	19	24	25	27	29	28	26	27	25	25
extracted points	19	19	19	18	22	15	17	16	18	18	17	16	19	19	18
new points	2	1	3	5	1	7	1	4	2	3	3	2	4	5	2

5.5.4 Martin Marietta R3 Sequence

The last sequence is one of the four sequences distributed by Martin Marietta as part of the UGV-RSTA project. As in the third sequence, the camera is mounted on a vehicle and the images are taken when the vehicle is moving through an outdoor environment. The original sequence consists of densely sampled images of size 347 × 238; only thirty frames (every fifth frame) in the original sequence were used in the experiment. During the acquisition of the images, the vehicle moves to the right and slightly into the scene. Figure 5.8 shows the trajectories of a set of feature points from the first frame to the 19th and 30th frames. As seen from the figures, the points on the mountain are far away from the vehicle, resulting in small movements on the image plane. Figure 5.8 also shows the feature points detected in the 2nd, 19th and 30th frames and the points added to the tracking list. The dynamic inclusion of the new feature points is summarized in Table 5.4.

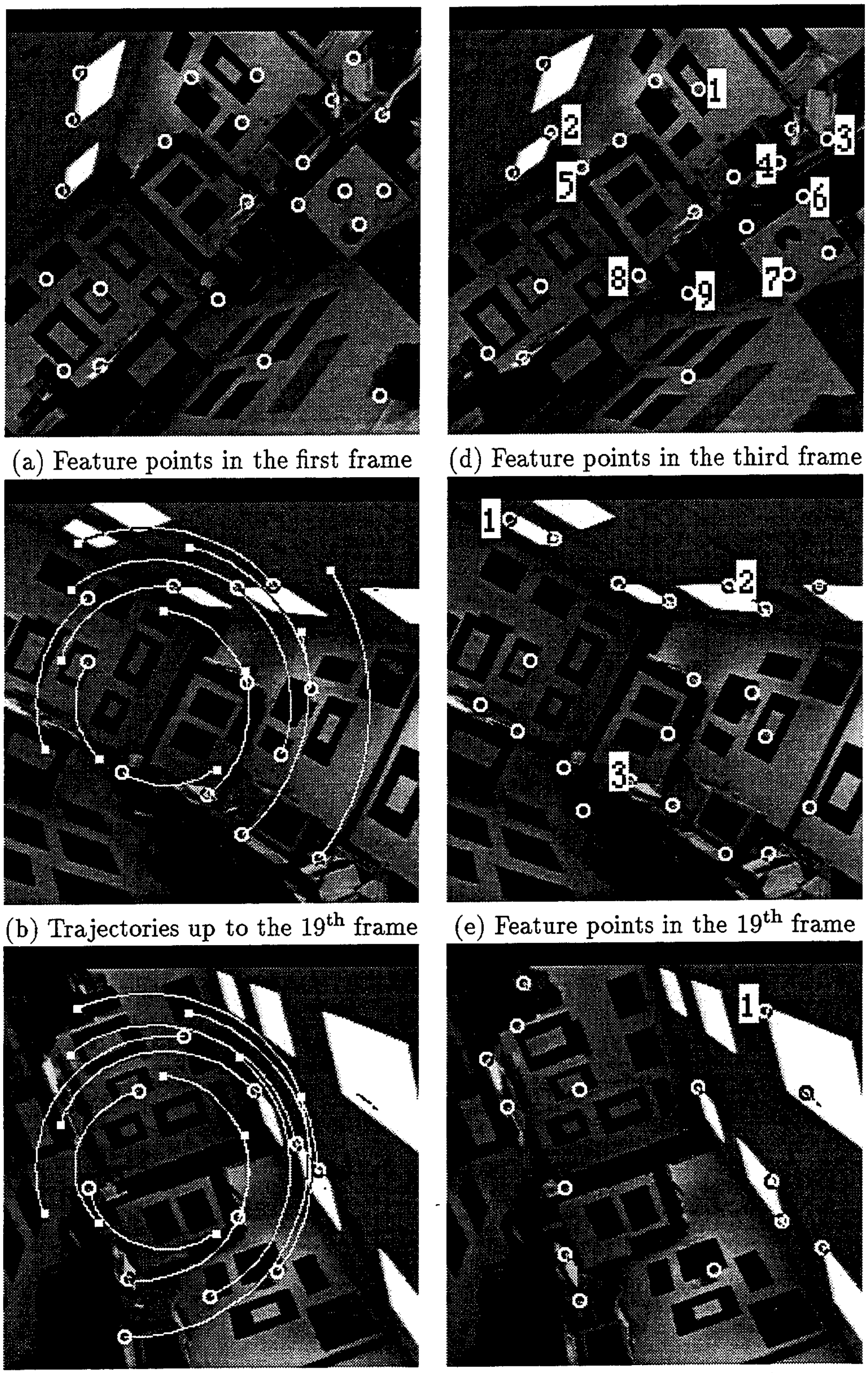
Table 5.4: The number of feature points in the tracking list for the Martin Marietta R3 Sequence

frame number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
tracked points	0	9	12	18	21	20	23	22	25	25	27	27	30	30	28
extracted points	9	15	17	18	15	15	13	14	10	17	11	15	17	10	12
new points	9	4	8	3	0	3	0	3	1	4	0	3	2	0	2

frame number	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
tracked points	27	29	30	31	32	33	33	33	37	38	37	37	37	39	37
extracted points	18	19	18	11	15	17	15	17	19	19	20	15	16	10	17
new points	2	1	4	1	2	2	1	4	1	2	3	3	5	0	3

5.6 Conclusions

This chapter has presented an algorithm for tracking a dynamic set of feature points to sub-pixel accuracy over a sequence of images. To exploit the temporal information contained in a sequence, a simple 2D kinematic motion model is employed **locally** to describe the trajectory of each feature point. Due to the use of a localized tracking scheme, complicated 3D motion/structure estimation problems are avoided. To account for non-smooth changes in the image motion arising from the 3D movements of the camera, an inter-frame motion estimation scheme is designed. Therefore, the algorithm is able to follow arbitrary movements of feature points. Moreover, a scheme which is able to include as well as track new points detected in the subsequent frames is proposed. The scheme efficiently preserves the information in a sequence, and thus makes the algorithm useful for estimating the pose and ego-motion of the camera.



(a) Feature points in the first frame

(d) Feature points in the third frame

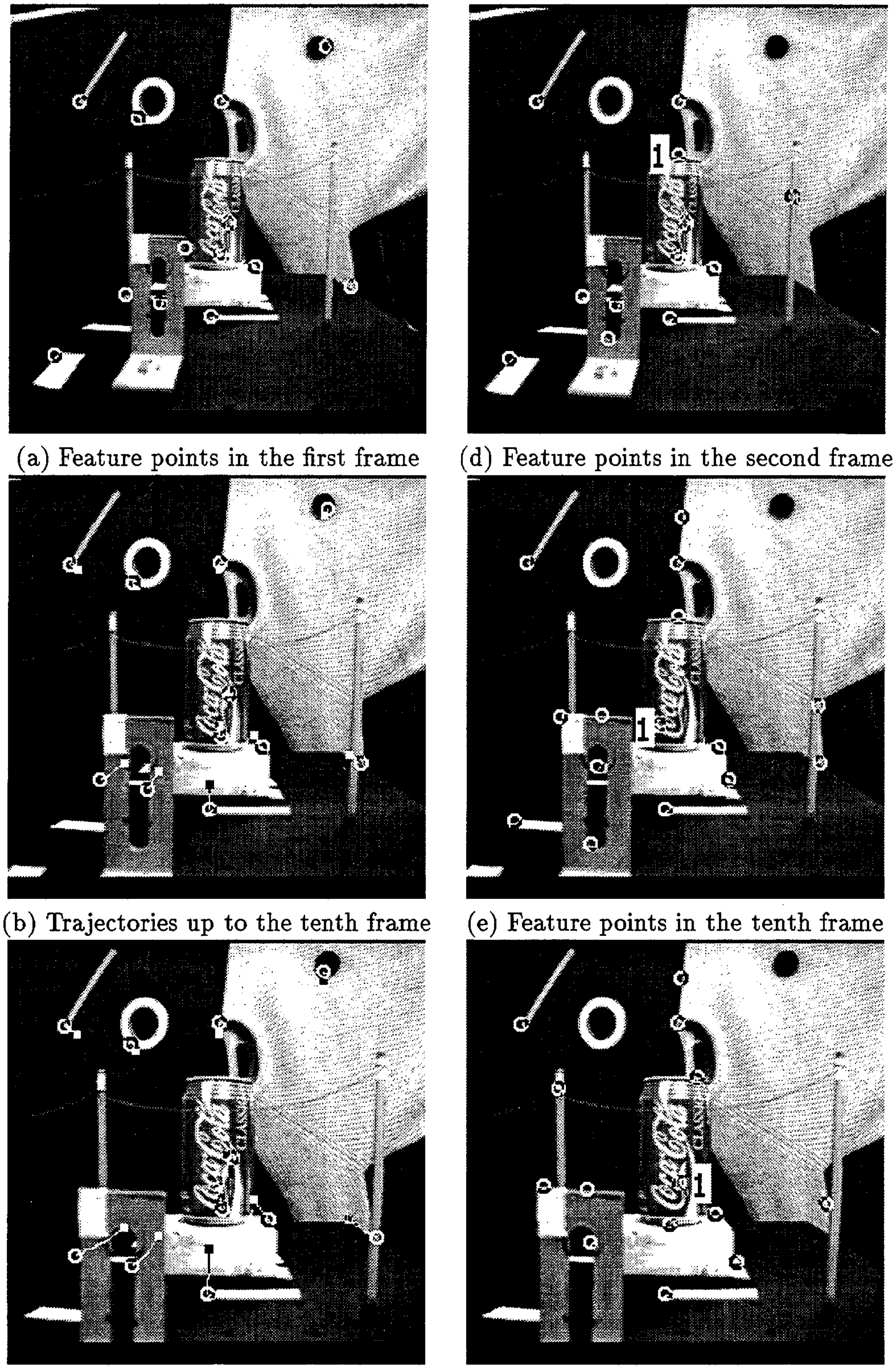
(b) Trajectories up to the 19th frame

(e) Feature points in the 19th frame

(c) Trajectories up to the 30th frame

(f) Feature points in the 30th frame

Figure 5.5: Tracking results for the UMASS PUMA2 Sequence



(a) Feature points in the first frame

(d) Feature points in the second frame

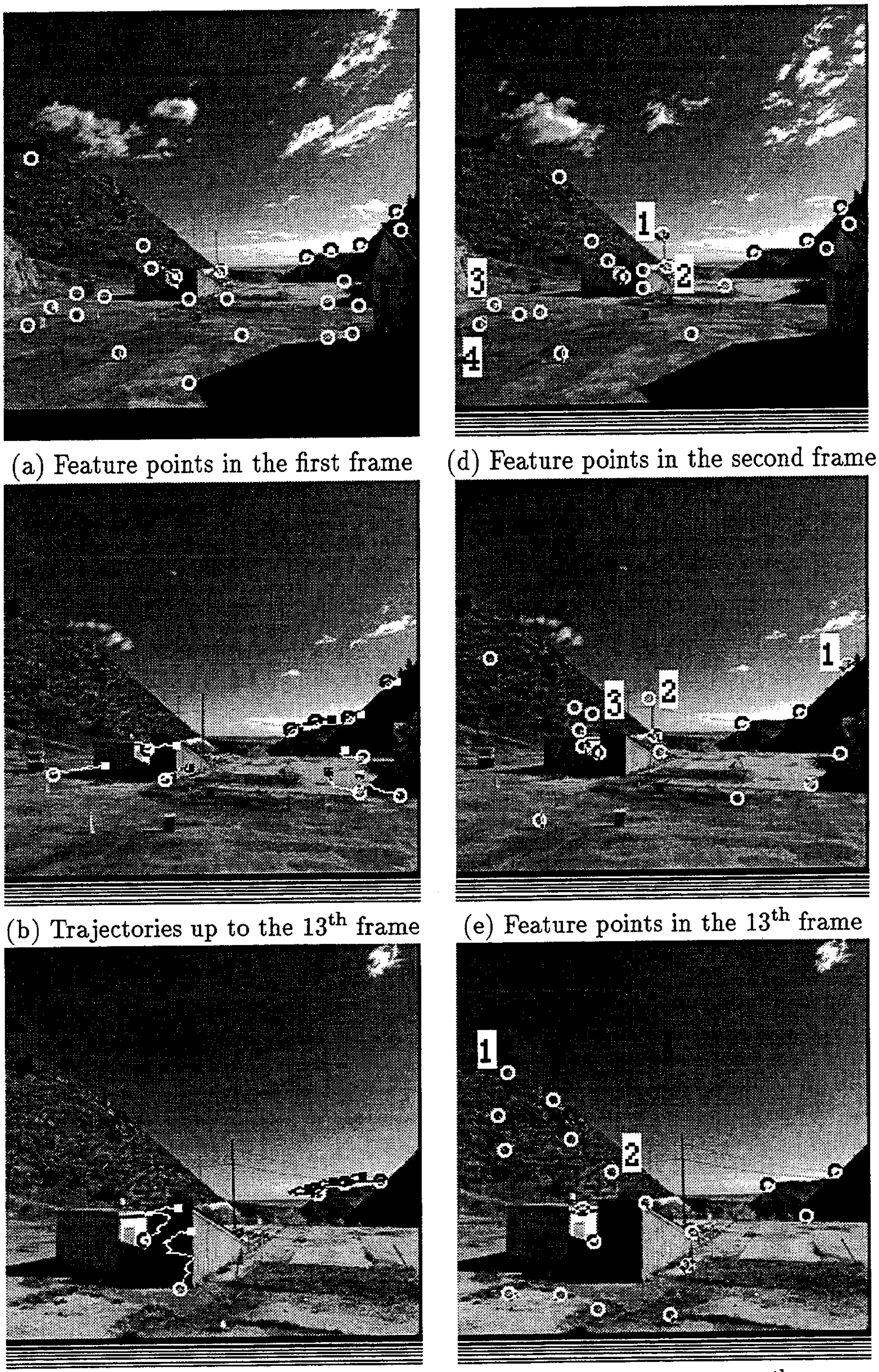
(b) Trajectories up to the tenth frame

(e) Feature points in the tenth frame

(c) Trajectories up to the 15th frame

(f) Feature points in the 15th frame

Figure 5.6: Tracking results for the Coke Can Sequence



(a) Feature points in the first frame

(d) Feature points in the second frame

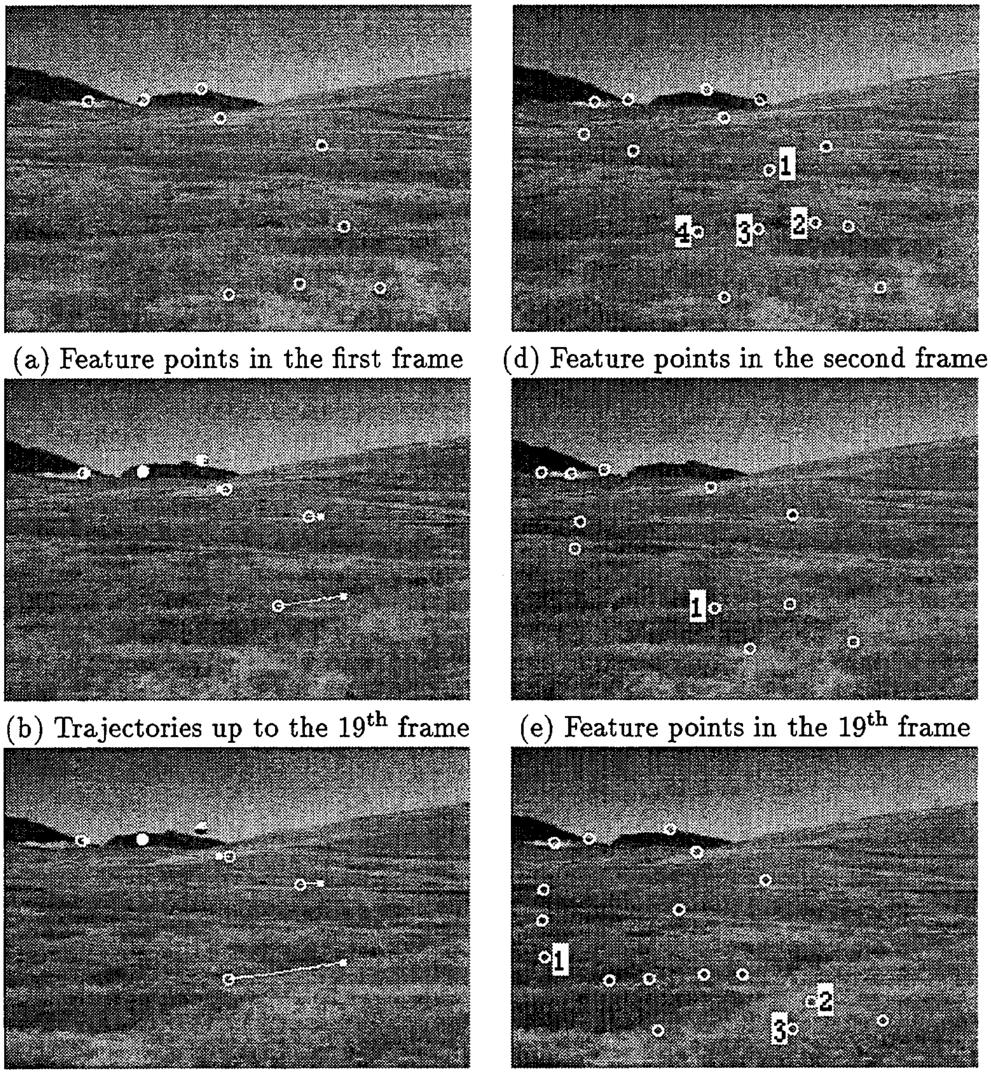
(b) Trajectories up to the 13th frame

(e) Feature points in the 13th frame

(c) Trajectories up to the 30th frame

(f) Feature points in the 30th frame

Figure 5.7: Tracking results for the Rocket ALV Sequence



(a) Feature points in the first frame

(d) Feature points in the second frame

(b) Trajectories up to the 19th frame

(e) Feature points in the 19th frame

(c) Trajectories up to the 30th frame

(f) Feature points in the 30th frame

Figure 5.8: Tracking results for the Martin Marietta R3 Sequence

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation, several image sequence analysis problems important for off-road vehicular navigation have been studied.

We first study the issue of image stabilization, i.e. the removal of unwanted image motion resulting from camera rotation. Multiple visual cues are integrated and the dynamic nature of a sequence is exploited using EKF in both calibrated and uncalibrated stabilization schemes. In particular, in the calibrated scheme, distant points and horizon lines are integrated by applying the same imaging model to the respective 3D position vector and normal vector. On the other hand, for the uncalibrated sequences, the imaging model cannot be applied to horizon lines detected from the images. Alternatively, the image motion of horizon lines is characterized by the motion of the associated 2D image plane normal vector. This results in different projective transformations in describing the image motion of distant points and horizon lines. However, since the corresponding projective coefficients are not sensitive to the variations of intrinsic parameters, only three rotational parameters need to be estimated in the uncalibrated scheme, as in the calibrated scheme. Consequently, points and lines are integrated in both stabilization schemes. Due to the direct estimation of parameters relevant to stabilization, the approach is therefore simple and robust. Most importantly, it is suitable for subsequent motion analysis since the residual motion in the stabilized sequence has been analyzed.

Subsequently, we investigate the recovery of motion/structure from selective stabilization. The algorithm first uses the proposed image stabilization scheme for the estimation of total rotation. Then, to separate smooth rotation and residual rotation, a kinetic model of the platform hosting the camera is employed. By applying a maneuver detection scheme for the detection of the beginning and end of smooth rotation, appropriate dynamic laws are thereafter used to achieve selective stabilization. Consequently, the structure parameters of a set of close features are estimated in a less perturbed frame of reference. Due to the incorporation of kinetic models, the proposed scheme is therefore more closely related to mechanical stabilization, and provides an alternative solution to the problem.

Finally, we study the problem of feature correspondence over a sequence. A 2D, localized feature point tracking algorithm is proposed. The algorithm employs a simple, 2D kinematic model to describe the trajectory of a feature point. Feature correspondences are obtained by recursively applying the following three steps: inter-frame motion estimation, corresponding point identification and temporal information processing. The inter-frame motion estimation step compensates for the motion between two windows likely to contain the matching points. The step of corresponding point identification consists of several procedures such as correlation matching and interpolation to identify the matching point to sub-pixel accuracy. The last step incorporates the information contained in the current frame for further tracking. Due to the localized operations of the algorithm, the scheme is able to account for the effect of 3D camera motion. In addition, we also consider the inclusion of new feature points detected from subsequent frames. An efficient scheme to maintain the number of points being tracked is designed. Moreover, since each feature point is tracked independently, parallel implementations of the proposed algorithm are possible.

6.2 Suggestions for Future Work

In this final section, we suggest a few extensions of this dissertation.

- For image stabilization, we have implicitly assumed that distant features can be differentiated from nearby features by active sensors such as laser radars. This leads to a sensor fusion problem. Alternatively, such problems can be approached from an image understanding point of view. A batch scheme which processes the first few images may be designed to select desired features.

- Currently, we have tested the image stabilization algorithm on the images taken from a camera mounted on a moving vehicle. **Distant** features are assumed to exist in the images. However, the algorithm should be tried on images where the scene is **close** to the camera. In these images, if there exist features whose image motion is mainly due to the rotation, they can be used to estimate the rotational parameters for achieving image stabilization.

- For selective stabilization, the kinetic model of a vehicle has been used in this dissertation to model the residual oscillatory behavior. Other approaches may be employed. For example, the parameters which best describe the oscillatory behavior of the host can be estimated from the image motion of distant features in the first few seconds. The resulting model can then be used for achieving selective stabilization.

- Feedback control and mechanical stabilization. The separation of oscillatory rotation and smooth rotation provides useful information for the design of a suspension system. It can also be combined with inertial data measured from inertial sensors for achieving mechanical stabilization.

- Model-based compression. Since image stabilization results in a translation-only sequence, model-based compression schemes can be developed. For example, an affine transformation can be used to model the motion in a stabilized sequence. The scaling parameter can be employed to account for the looming component, while the 2D translational parameters can take into account the other two degrees of freedom.

- Hardware implementation. The image stabilization and feature tracking algorithms now run on Sun SPARC stations. Since many computationally expensive operations such as matrix computations are required for these schemes, real-time implementations of the algorithms will be challenging tasks. Appropriate modifications may be required in order to use image processing hardware such as the Datacube, or a DSP chip like the C80, for real-time performance.

Appendix A

Derivation of Image Motion

We give the derivation of image motion in vector form in this appendix. As in Section 3.2, assume that perspective projection is used as the imaging model and the camera undergoes translation and rotation. Then, for a 3D point $\mathbf{P} : (X, Y, Z)^T$, the image motion of the projection point $\mathbf{p} : (x, y)^T$ can be obtained as follows:

$$\begin{aligned}
\dot{\mathbf{p}} &= \frac{\partial \mathcal{P}}{\partial \mathbf{P}} \dot{\mathbf{P}} + \dot{\mathbf{p}}_c && \text{(A.1)} \\
&= f_c \begin{bmatrix} \frac{1}{Z} & 0 & -\frac{X}{Z^2} \\ 0 & \frac{1}{Z} & -\frac{Y}{Z^2} \end{bmatrix} \dot{\mathbf{P}} && \text{(A.2)} \\
&= \frac{1}{Z} \begin{bmatrix} f_c & 0 & -(x - x_c) \\ 0 & f_c & -(y - y_c) \end{bmatrix} (-\boldsymbol{\omega} \times \mathbf{P} - \mathbf{V}) && \text{(A.3)} \\
&= f_c^{-1} \begin{bmatrix} f_c & 0 & -(x - x_c) \\ 0 & f_c & -(y - y_c) \end{bmatrix} \left\{ \begin{bmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \\ f_c \end{bmatrix} - \begin{bmatrix} f_c \frac{V_x}{V_z} \\ f_c \frac{V_y}{V_z} \\ f_c \end{bmatrix} \frac{V_z}{Z} \right\}
\end{aligned}$$

As before, define $\mathbf{p}_{\text{foe}} : (f_c \frac{V_x}{V_z}, f_c \frac{V_y}{V_z})^T + \mathbf{p}_c$ and $\tau : \frac{Z}{V_z}$ as well as the following matrices:

$$\boldsymbol{\Omega}_z = \begin{bmatrix} 0 & \omega_z \\ -\omega_z & 0 \end{bmatrix} \quad \boldsymbol{\omega}_{xy}^{\perp} = \begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix} \tag{A.4}$$

Then (A.4) can be rewritten as

$$\dot{\mathbf{p}} = f_c^{-1} \left[\begin{array}{c|c} f_c \mathbf{I} & -(\mathbf{p} - \mathbf{p}_c) \end{array} \right] \left\{ \left[\begin{array}{c|c} \boldsymbol{\Omega}_z & \boldsymbol{\omega}_{xy}^{\perp} \\ - & - \\ -\boldsymbol{\omega}_{xy}^{\perp T} & 0 \end{array} \right] \begin{bmatrix} \mathbf{p} - \mathbf{p}_c \\ - \\ f_c \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{\text{foe}} - \mathbf{p}_c \\ - \\ f_c \end{bmatrix} \frac{1}{\tau} \right\} \tag{A.5}$$

Consequently, we have

$$\dot{\mathbf{p}} = f_c^{-1} (\mathbf{p} - \mathbf{p}_c) \boldsymbol{\omega}_{xy}^{\perp T} (\mathbf{p} - \mathbf{p}_c) + \boldsymbol{\Omega}_z (\mathbf{p} - \mathbf{p}_c) + f_c \boldsymbol{\omega}_{xy}^{\perp} - (\mathbf{p}_{\text{foe}} - \mathbf{p}) \frac{1}{\tau} \tag{A.6}$$

Appendix B

Representation of the Image Plane Normal Vector

We derive the relationship between $\mathbf{w}$, the projection of the 3D normal vector of the plane Π (see Figure 3.2), and the image plane normal $\mathbf{o}$ in this appendix. Consider an image horizon line denoted by

$$\mathcal{L} : y = a + bx \tag{B.1}$$

where $(x, y)^T$ are the image plane coordinates of a 3D point. Then the image plane vector normal to $\mathcal{L}$, defined in (3.20), is

$$\mathbf{o} = (-\frac{b}{a}, \frac{1}{a})^T \tag{B.2}$$

Let f_c be the focal length of the camera and assume that the optical axis intersects the image plane at $\mathbf{p}_c : (x_c, y_c)^T$. Then

$$\begin{array}{rcl} \mathbf{P}_1 & : & (0, a + bx_c - y_c, f_c)^T \\ \mathbf{P}_2 & : & (\frac{y_c - a - bx_c}{b}, 0, f_c)^T \end{array} \tag{B.3}$$

are the coordinates of two points lying on the plane Π, with respect to the camera frame of reference. Consequently, by taking the cross product of $\mathbf{P}_1$ and $\mathbf{P}_2$, we have

$$\mathbf{N} \equiv \mathbf{P}_1 \times \mathbf{P}_2 \tag{B.4}$$

$$= (1, -\frac{1}{b}, \frac{a + bx_c - y_c}{bf_c})^T \tag{B.5}$$

The 3D normal vector $\mathbf{W}$ is then obtained by

$$\mathbf{W} = \frac{\mathbf{N}}{||\mathbf{N}||} \tag{B.6}$$

$\mathbf{w}$ is related to $\mathbf{o}$ by

$$\mathbf{w} = \mathcal{P}(\mathbf{W}) + \mathbf{p}_c \tag{B.7}$$

$$= \left[\begin{array}{c} f_c^2 \frac{b}{a + bx_c - y_c} \\ -f_c^2 \frac{1}{a + bx_c - y_c} \end{array} \right] + \mathbf{p}_c \tag{B.8}$$

$$= -\frac{f_c^2}{1 - < \mathbf{p}_c, \mathbf{o} >} \mathbf{o} + \mathbf{p}_c \tag{B.9}$$

Appendix C

Derivation of Vehicle Dynamics

We apply Lagrange's equations of motion to derive the residual oscillatory behavior of the four-wheel vehicle model in this appendix.

Consider Figures 3.3 and 3.4. Define the generalized coordinates, consisting of seven degrees of freedom, as

$$\mathbf{q} \equiv (x_1, x_3, x_5, x_7, x_c, \theta, \phi)^T \tag{C.1}$$

Then Lagrange's equations of motion are written as follows [16]:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_j}\right) - \frac{\partial T}{\partial q_j} + \frac{\partial U}{\partial q_j} + \frac{\partial D}{\partial \dot{q}_j} = Q_j \quad j = 1, \ldots, 7 \tag{C.2}$$

where $\{q_j, j = 1, \ldots, 7\}$ constitute the generalized coordinates, T is the kinetic energy of the system, U is the potential energy, D is the dissipation function and the Q_j's represent the generalized forces acting on the system during vibration. In particular, as regards the behavior of $x_c, \theta, \dot{\theta}$, if the suspension system is of passive type, then the corresponding generalized force is assumed to be zero. We look at various energy terms in the following paragraphs.

- Kinetic energy

It is well known that the kinetic energy of a moving rigid body is equal to the sum of the translational and rotational kinetic energy. Therefore, taking into account the movements of the unsprung elements and oscillations of the sprung element, the kinetic energy is

$$T = T_B + T_w \tag{C.3}$$

where T_B is the kinetic energy contributed by the sprung element

$$T_B = \frac{1}{2}M_B\dot{x}_c^2 + \frac{1}{2}I_{yy}\dot{\theta}^2 + \frac{1}{2}I_{zz}\dot{\phi}^2 \tag{C.4}$$

with M_B denoting the mass of the sprung element, and I_{yy} and I_{zz} are the moments of inertia with respect to the pitch and roll axes. T_w, on the other hand, is the kinetic energy contributed by the unsprung elements:

$$T_w = \frac{1}{2}M_{wf}\dot{x}_1^2 + \frac{1}{2}M_{wf}\dot{x}_3^2 + \frac{1}{2}M_{wr}\dot{x}_5^2 + \frac{1}{2}M_{wr}\dot{x}_7^2 \tag{C.5}$$

- Potential energy

The potential energy U takes into account the influences of suspension springs and tires on a vehicle undergoing vibration. To obtain U, the deformations of suspension springs are first computed. They are approximated by

$$\begin{aligned} d_{f,r} &\approx x_c - T_{s,r}\phi + W_A\theta - x_1 \\ d_{f,l} &\approx x_c + T_{s,l}\phi + W_A\theta - x_3 \\ d_{r,r} &\approx x_c - T_{s,r}\phi - W_B\theta - x_5 \\ d_{r,l} &\approx x_c + T_{s,l}\phi - W_B\theta - x_7 \end{aligned} \tag{C.6}$$

Then assuming that linear relationships hold between the deformations and the forces generated by the springs, the potential energy is

$$U = U_S + U_T \tag{C.7}$$

where U_S is the energy stored in the suspension systems

$$U_S = \frac{1}{2}K_{f,r}d_{f,r}^2 + \frac{1}{2}K_{f,l}d_{f,l}^2 + \frac{1}{2}K_{r,r}d_{r,f}^2 + \frac{1}{2}K_{r,l}d_{r,r} \tag{C.8}$$

and U_T is the energy stored in the tires,

$$U_T = \frac{1}{2}K_T\left[(x_1 - x_{01})^2 + (x_3 - x_{02})^2 + (x_5 - x_{03})^2 + (x_7 - x_{04})^2\right] \tag{C.9}$$

- Dissipation function

The dissipation function accounts for the effects of the shock absorbers. Assuming that the forces generated by the shock absorbers vary linearly with the deforming rate, the dissipation function D is

$$D = \frac{1}{2}C_{f,r}\dot{d}_{fr}^2 + \frac{1}{2}C_{f,l}\dot{d}_{fl}^2 + \frac{1}{2}C_{r,r}\dot{d}_{rr}^2 + \frac{1}{2}C_{r,l}\dot{d}_{rl}^2 \tag{C.10}$$

After various energy terms have been considered, (C.2) can be applied. The residual oscillatory behavior, especially θ and ϕ, of the four-wheel vehicle model is obtained.

Appendix D

Performance Analysis of Approximate Kinetic Laws

In this appendix, we analytically compare the performance of the approximate dynamic law of pitch and roll motion in (4.3) to the more complete dynamic law in (3.34). Equivalently, assume that the smooth rotation and the residual oscillatory rotation are described by

$$\dot{\mathbf{x}}_0 = \mathbf{G}\mathbf{x}_0 + \mathbf{\Gamma}\mathbf{u}_{us} \tag{D.1}$$

where $\mathbf{x}_0 = (\omega_{sy}, \omega_{sz}, \mathbf{x}_{us}^T)^T$, $\mathbf{G}$ is defined in (4.13), and

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 10} \\ \mathbf{0}_{4\times 2} & \mathbf{\Gamma}_{us} \end{bmatrix} \tag{D.2}$$

with $\mathbf{\Gamma}_{us}$ mentioned in (3.34). We study the behavior of the following linear systems:

$$\begin{cases} \dot{\mathbf{x}}_1 &= \mathbf{G}\mathbf{x}_1 \\ \omega_1 &= \mathbf{F}\mathbf{x}_1 \end{cases} \tag{D.3}$$

where $\mathbf{x}_1$ consists of the same parameters as $\mathbf{x}_0$. ω_1 denotes the measurable total angular velocity, as defined in (4.12).

We employ a prediction-correction based scheme to separate the smooth and residual oscillatory rotation [12]:

$$\dot{\hat{\mathbf{x}}}_1 = \mathbf{G}\hat{\mathbf{x}}_1 + \mathbf{K}_1[\omega_1 - \mathbf{F}\hat{\mathbf{x}}_1] \tag{D.4}$$

The resulting error $\tilde{\mathbf{x}}_1 = \mathbf{x}_0 - \hat{\mathbf{x}}_1$ then satisfies

$$\dot{\tilde{\mathbf{x}}}_1 = (\mathbf{G} - \mathbf{K}_1\mathbf{F})\tilde{\mathbf{x}}_1 + \mathbf{\Gamma}\mathbf{u}_{us} \tag{D.5}$$

By choosing an appropriate $\mathbf{K}_1$, $\tilde{\mathbf{x}}_1$ will remain bounded as long as $\mathbf{u}_{us}$ is bounded. Therefore, given reliable estimates of the total angular velocity, we can employ the approximate dynamic law (D.3) to achieve selective stabilization without much degradation.

Acknowledgements

First of all, I wish to thank my advisor, Dr. Rama Chellappa, for his endless encouragement and support throughout the course of this work. With his profound insight and technical knowledge, he has made this dissertation possible. Many thanks are also due to the other members of the committee, Drs. Larry Davis, KuoJuey Ray Liu, Prakash Narayan, and Azriel Rosenfeld for their kind advice.

Special thanks go to Drs. Ting-Hu Wu and Philippe Burlina for their invaluable help and friendship as well as to Dr. Gem-Sun Young, who introduced me into the field of visual motion analysis. I would also like to thank all my colleagues in the Center for Automation Research (CfAR), especially Arnab Das, Dr. Santhana Krishnamachari, Dr. Oh-Jin Kwon, Shyam Kuttikkad, Chih-lung Lin, Reuven Meth, Carlos Morimoto, Rahul Parulekar, Saad Sirohey, and Murari Srinivasan, who have made my years at Maryland memorable. Finally, thanks are due to all the staff in CfAR for their kindness and assistance, which has made CfAR an enjoyable workplace.

Bibliography

[1] G. Adiv, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 7, pp. 384–401, July 1985.

[2] J.K. Aggarwal, "Motion and Time-Varying Imagery—An Overview," in *Proc. of IEEE Workshop on Motion: Representation and Analysis* (Kiawah Island, SC), pp. 1–6, May 1986.

[3] J.Y. Aloimonos, Z. Duric, C. Fermuller, L. Huang, E. Rivlin, and R. Sharma, "Behavioral Visual Motion Analysis," in *Proc. of DARPA Image Understanding Workshop*, pp. 521–533, Jan. 1992.

[4] D.H. Ballard and O.A. Kimball, "Rigid Body Motion from Depth and Optical Flow," *Computer Vision, Graphics, and Image Processing*, Vol. 22, pp. 95–115, 1983.

[5] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, San Diego, CA: Academic Press, 1988.

[6] S.D. Blostein and T.S. Huang, "Detecting Small, Moving Objects in Image Sequences Using Sequential Hypothesis Testing," *IEEE Trans. on Signal Processing*, Vol. 39, pp. 1611–1629, July 1991.

[7] T.J. Broida and R. Chellappa, "Estimating the Kinematics and Structure of a Rigid Object from a Sequence of Monocular Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, pp. 497–513, June 1991.

[8] L.G. Brown, "A Survey of Image Registration Techniques," *ACM Computing Servey*, Vol. 24, pp. 325–376, Dec. 1992.

[9] P. Burlina and R. Chellappa, "Time-to-X: Analysis of Motion through Temporal Parameters," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition* (Seattle, WA), pp. 461–468, June 1994.

[10] P. Burt and P. Anandan, "Image Stabilization by Registration to a Reference Mosaic," in *Proc. of ARPA Image Understanding Workshop* (Monterey, CA), pp. 425–434, Nov. 1994.

[11] Y.L. Chang and J.K. Aggarwal, "3D Structure Reconstruction From an Ego Motion Sequence Using Statistical Estimation and Detection Theory," in *Proc. of IEEE Workshop on Visual Motion* (Princeton, NJ), pp. 268–273, Oct. 1991.

[12] C.T. Chen, *Linear System Theory and Design*, New York: Holt, Rinehart, and Winston, 1984.

[13] I.J. Cox, "A Review of Statistical Data Association Techniques for Motion Correspondence," *International Journal of Computer Vision*, Vol. 10, pp. 53–66, Aug. 1993.

[14] N. Cui, J. Weng, and P. Cohen, "Extended Structure and Motion Analysis from Monocular image Sequences," in *Proc. of International Conf. on Computer Vision* (Osaka, Japan), pp. 222–229, Dec. 1990.

[15] L.S. Davis, R. Bacjsy, R. Nelson, and M. Herman, "RSTA on the Move," in *Proc. of ARPA Image Understanding Workshop* (Monterey, CA), pp. 435–456, Nov. 1994.

[16] E.A. Desloge, *Classical Mechanics*, vol. 2, New York:John Wiley & Sons, 1982.

[17] E.D. Dickmanns and V. Graefe, "Dynamic Monocular Machine Vision," *Machine Vision and Applications*, Vol. 1, pp. 223–240, 1988.

[18] E.D. Dickmanns and B.D. Mysliwetz, "Recursive 3-D Road and Relative Ego-State Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, pp. 199–213, Feb. 1992.

[19] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.

[20] J.Q. Fang and T.S. Huang, "Some Experiments on Estimating the 3-D Motion Parameters of a Rigid Body From Two Consecutive Image Frames," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 545–554, Sept. 1984.

[21] C. Fermüller and Y. Aloimonos, "Qualitative Egomotion," *International Journal of Computer Vision*, Vol. 15, pp. 7–29, June 1995.

[22] A. Hady and D.A. Crolla, "Theoretical Analysis of Active Suspension Performance Using a Four-wheel Vehicle Model," *Proc. of the Institution of Mechanical Engineers*, Vol. 203, pp. 125–135, 1989.

[23] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt, "Real-time Scene Stabilization and Mosaic Construction," in *Proc. of ARPA Image Understanding Workshop* (Monterey, CA), pp. 457–465, Nov. 1994.

[24] D.J. Heeger, "Optical Flow Using Spatiotemporal Filters," *International Journal of Computer Vision*, Vol. 1, pp. 279–302, 1988.

[25] J. Heel, "Direct Estimation of Structure and Motion from Multiple Frames," AI Memo 1190, MIT Artificial Intelligence Laboratory, March 1990.

[26] B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," *Artifical Intelligence*, Vol. 17, pp. 185–203, Aug. 1981.

[27] T.S. Huang, ed., *Image Sequence Analysis*, Berlin: Springer-Verlag, 1981.

[28] T.S. Huang *et al.*, "Motion Detection and Estimation from Stereo Image Sequences: Some Preliminary Experimental Results," in *Proc. of IEEE Workshop on Motion: Representation and Analysis* (Kiawah Island, SC), pp. 45–46, May 1986.

[29] M. Irani, B. Rousso, and S. Peleg, "Recovery of Ego-Motion Using Image Stabilization," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition* (Seattle, WA), pp. 454–460, June 1994.

[30] J. Kearney, W. Thompson, and D. Boley, "Optical Flow Estimation : An Error Analysis of Gradient-Based Methods with Local Optimization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 9, pp. 229–244, March 1987.

[31] R. Kumar, P. Anandan, and K. Hanna, "Shape Recovery from Multiple Views: A Parallax Based Approach," in *Proc. of ARPA Image Understanding Workshop* (Monterey, CA), pp. 947–956, Nov. 1994.

[32] O.J. Kwon, R. Chellappa, and C. Morimoto, "Motion Compensated Subband Coding of Video Acquired from a Moving Platform," in *Proc. of IEEE International Conf. on Acoustics, Speech, and Signal Processing* (Detroit,MI), 1995.

[33] Y. Liu and T.S. Huang, "Vehicle-Type Motion Estimation from Multi-frame Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, pp. 802–808, Aug. 1993.

[34] H.C. Longuet-Higgins, "The Reconstruction of a Scene From Two Projections—Configurations that Defeat the 8-point Algorithm," in *Proc. of IEEE Conf. on Artificial Intelligence Applications* (Denver, CO), pp. 395–397, Dec. 1984.

[35] H.C. Longuet-Higgins and K. Prazdny, "The Interpretation of a Moving Retinal Image," *Proc. Royal Society London B*, Vol. 208, pp. 385–397, 1980.

[36] B.S. Manjunath, R. Chellappa, and C.V. Malsburg, "A Feature Based Approach to Face Recognition," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition* (Champaign, IL), pp. 373–378, June 1992.

[37] P. Michelberger, L. Palkovics, and J. Bokor, "Robust Design of Active Suspension System," *International Journal of Vehicle Design*, Vol. 14, pp. 145–165, 1993.

[38] C. Morimoto, D. DeMenthon, L. Davis, R. Chellappa, and R. Nelson, "Detection of Independently Moving Objects in Passive Video," in *Proc. of Intelligent Vehicles Symposium* (Detroit, MI), Sept. 1995.

[39] H.H. Nagel and W. Enkelmann, "An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields From Image Sequences," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 565–593, 1986.

[40] S. Negahdaripour and B.K.P. Horn, "Direct Passive Navigation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 9, pp. 168–176, Jan. 1987.

[41] R. Nelson, "Qualitative Detection of Motion by a Moving Observer," *International Journal of Computer Vision*, Vol. 7, pp. 33–46, 1991.

[42] T. Poggio, A. Verri, and V. Torre, "Green Theorems and Qualitative Properties of the Optical Flow," AI Memo 1289, MIT Artificial Intelligence Laboratory, April 1991.

[43] J. Roach and J. Aggarwal, "Determining the Movement of Objects from a Sequence of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 2, pp. 554–562, Nov. 1980.

[44] F.R. Schell and E.D. Dickmanns, "Autonomous Landing of Airplanes by Dynamic Machine Vision," *Machine Vision and Applications*, Vol. 7, pp. 127–134, 1994.

[45] I.K. Sethi and R. Jain, "Finding Trajectories of Feature Points in a Monocular Image Sequence," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 9, pp. 56–73, Jan. 1987.

[46] H. Shariat and K. Price, "Motion Estimation with More Than Two Frames," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 417–434, May 1990.

[47] D. Sinclair, A. Blake, and D. Murray, "Robust Estimation of Egomotion from Normal Flow," *International Journal of Computer Vision*, Vol. 13, pp. 57–69, Sept. 1994.

[48] M. Subbarao, "Bounds on Time-to-Collision and Rotational Component from First Order Derivatives of Image Flow," *Computer Vision, Graphics and Image Processing*, Vol. 50, pp. 329–341, June 1990.

[49] W.B. Thompson, K.M. Mutch, and V.A. Berzins, "Dynamic Occlusion Analysis in Optical Flow Fields," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 7, pp. 374–383, July 1985.

[50] R.Y. Tsai and T.S. Huang, "Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 29, pp. 1147–1152, Dec. 1981.

[51] R.Y. Tsai and T.S. Huang, "Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch III : Finite Point Correspondences and the Three View Problem," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 32, pp. 213–220, Apr. 1984.

[52] R.Y. Tsai, T.S. Huang, and W.L. Zhu, "Estimating Three-Dimensional Motion Parameters of a Rigid Plananr Patch, II: Singular Value Decomposition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. 30, pp. 525–534, Aug. 1982.

[53] T. Vieville, E. Clergue, and P. Facao, "Computation of Ego-Motion and Structure from Visual and Internal Sensors Using the Vertical Cue," in *Proc. of International Conf. on Computer Vision* (Berlin, Germany), pp. 591–598, May 1993.

[54] A.M. Waxman, B. Kamgar-Parsi, and M. Subbarao, "Closed-Form Solutions to Image Flow Equations for 3D Structure and Motion," *International Journal of Computer Vision*, Vol. 1, pp. 239–258, 1987.

[55] J. Weng, N. Ahuja, and T.S. Huang, "Matching Two Perspective Views," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, pp. 806–825, Aug. 1992.

[56] J. Weng, T.S. Huang, and N. Ahuja, "3-D Motion Estimation, Understanding, and Prediction from Noisy Image Sequences," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 9, pp. 370–389, May 1987.

[57] G. Wolberg, *Digital Image Warping*, Los Alamitos, CA: IEEE Computer Society Press, 1990.

[58] T.H. Wu, R. Chellappa, and Q. Zheng, "Experiments on Estimating Egomotion and Structure Parameters Using Long Monocular Image Sequences," *International Journal of Computer Vision*, Vol. 15, pp. 77–103, June 1995.

[59] G.S. Young and R. Chellappa, "3-D Motion Estimation Using a Sequence of Noisy Stereo Images: Models, Estimation, and Uniqueness Results," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 735–759, Aug. 1990.

[60] Z. Zhang and O.D. Faugeras, "Three-Dimensional Motion Computation and Object Segmentation in a Long Sequence of Stereo Frames," *International Journal of Computer Vision*, Vol. 7, pp. 211–241, Aug. 1992.

[61] Q. Zheng and R. Chellappa, "Automatic Feature Point Extraction and Tracking in Image Sequences for Arbitrary Camera Motion," *International Journal of Computer Vision*, Vol. 15, pp. 31–76, June 1995.

[62] Y.J. Zheng, W. Ritter, and R. Jansen, "An Adaptive System for Traffic Sign Recognition," in *Proc. of Intelligent Vehicles Symposium* (Paris, France), pp. 165–170, 1994.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1995

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
Electronic Stabilization and Feature Tracking in Long Image Sequences

5. FUNDING NUMBERS
DAAH04-93-G-0419

6. AUTHOR(S)
Y.S. Yao and R. Chellappa

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

8. PERFORMING ORGANIZATION REPORT NUMBER
CAR-TR-790
CS-TR-3527

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ARO 32365.6-MA

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This dissertation is concerned with processing of visual motion with application to off-road vehicular navigation. We first consider the estimation of total rotation from an image sequence. We exploit the dynamic nature of the sequence and use multiple visual cues to perform image stabilization. Both calibrated and uncalibrated stabilization schemes are designed. The residual motion in a stabilized sequence is also analyzed. Next we address the issue of selective stabilization, defined as the separation of the smooth rotation and the residual vibrations caused by oscillatory rotation. We incorporate a kinetic model to explicitly account for vibration phenomena. A maneuver detection scheme, for detecting the beginning and end of smooth rotation, is designed to facilitate the selective stabilization. Finally, we study the problem of feature correspondence. We propose a localized feature point tracking algorithm, which employs a 2D kinematic model and relies on a Probabilistic Data Association Filter for the estimation of inter-frame motion. Corresponding points are identified to sub-pixel accuracy and an Extended Kalman Filter is employed to process the new data. The ability to dynamically include new feature points from subsequent frames also makes the algorithm suitable for structure from motion and tracking over a sequence.

14. SUBJECT TERMS
Image sequence stabilization, feature tracking, rotation estimation, vehicle motion

15. NUMBER OF PAGES
88

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements.***

Block 1. Agency Use Only *(Leave blank).*

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD -	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE -	See authorities.
NASA -	See Handbook NHB 2200.2.
NTIS -	Leave blank.

Block 12b. Distribution Code.

DOD -	Leave blank.
DOE -	Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA -	Leave blank.
NTIS -	Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only).*

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.